Curvilinear structure modeling and its applications in computer vision

Seong-Gyun Jeong

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS
ECOLE DOCTORAL STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis
Speciality : Computer Science

Defended by

Seong-Gyun JEONG

Curvilinear Structure Modeling and Its Applications in Computer Vision

Thesis Advisors: Josiane ZERUBIA and Yuliya TARABALKA
prepared at INRIA Sophia Antipolis, AYIN Team
defended on November 23, 2015

JURY

President :	Dr. Xavier DESCOMBES	-	INRIA (MORPHEME), France
Reviewers :	Prof. Christine GRAFFIGNE	-	Université Paris Descartes, France
	Prof. Tamás SZIRÁNYI	-	MTA SZTAKI & BME, Hungary
Examinator :	Prof. Pascal FUA	-	EPFL, Switzerland
Advisor :	Dr. Josiane ZERUBIA	-	INRIA (AYIN), France
Co-Advisor :	Dr. Yuliya TARABALKA	-	INRIA (TITANE), France

Abstract

In this dissertation, we propose curvilinear structure reconstruction models based on stochastic modeling and ranking learning system. We assume that the entire line network can be decomposed into a set of line segments with variable lengths and orientations. This assumption enables us to reconstruct arbitrary shapes of curvilinear structure for different types of datasets. We compute curvilinear feature descriptors based on the image gradient profiles and the morphological profiles. For the stochastic model, we propose prior constraints that define the spatial interaction of line segments. To obtain an optimal configuration corresponding to the latent curvilinear structure, we combine multiple line hypotheses which are computed by MCMC sampling with different parameter sets. Moreover, we learn a ranking function which predicts the correspondence of the given line segment and the latent curvilinear structures. A novel graph-based method is proposed to infer the underlying curvilinear structure using the output rankings of the line segments. We apply our models to analyze curvilinear structure on static images. Experimental results on wide types of datasets demonstrate that the proposed curvilinear structure modeling outperforms the state-of-the-art techniques.

Keywords: curvilinear structure modeling, marked point process, structure inference, machine learning

Résumé

Dans cette thèse, nous proposons des modèles de reconstruction de la structure curviligne fondée sur la modélisation stochastique et sur un système d'apprentissage structuré. Nous supposons que le réseau de lignes, dans sa totalité, peut être décomposé en un ensemble de segments de ligne avec des longueurs et orientations variables. Cette hypothèse nous permet de reconstituer des formes arbitraires de la structure curviligne pour différents types de jeux de données. Nous calculons les descripteurs des caractéristiques curvilignes fondés sur les profils des gradients d'image et les profils morphologiques. Pour le modèle stochastique, nous proposons des contraintes préalables qui définissent l'interaction spatiale des segments de ligne. Pour obtenir une configuration optimale correspondant á la structure curviligne latente, nous combinons plusieurs hypothèses de ligne qui sont calculées par échantillonnage MCMC avec différents jeux de paramètres. De plus, nous apprenons une fonction de classement qui prédit la correspondance du segment de ligne donné avec les structures curvilignes latentes. Une nouvelle méthode fondée sur les graphes est proposée afin d'inférer la structure sous-jacente curviligne en utilisant les classements de sortie des segments de ligne. Nous utilisons nos modèles pour analyser la structure curviligne sur des images statiques. Les résultats expérimentaux sur de nombreux types de jeux de données démontrent que les modèles de structure curviligne proposés surpassent les techniques de l'état de l'Art

Mots-clés: modélisation de structure curviligne, processus ponctuel marqué, inférence de structure, apprentissage automatique

Contents

Acronyms and Notations

Acronyms

ACO	Ant Colony Optimization
BD	Birth and Death
CC	Correlation Coefficient
CRF	Conditional Random Field
HMM	Hidden Markov Model
LT	Linear Transform
MAP	Mean Average Precision
MCMC	Markov chain Monte Carlo
MPP	Marked Point Process
MRF	Markov Random Field
NMS	Non Maximum Suppression
OOF	Optimally Oriented Flux
RJMCMC	Reversible jump Markov chain Monte Carlo
SA	Simulated Annealing
SSDT	Scale Space Distance Transform
SSVM	Structured Support Vector Machine
SVM	Support Vector Machine

General Notation

X	Random variable
x	Value of random variable (scalar)
$\mathbf{x}$	Value of random variable (vector)
$\mathcal{X}$	Vector space of the corresponding vector lives (*i.e.*, $x \in \mathcal{X}$ or $\mathbf{x} \in \mathcal{X}$)
$\Pr(X = x)$	Probability that random variable X taking a value x
$p(\cdot)$	Probability distribution

$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution of $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
λ	Eigenvalue
$\mathbb{R}^d$	d-dimensional real vector space
$\mathbb{N}$	Set of all natural numbers
$\hat{\cdot}$	Estimated value
$\lvert\cdot\rvert$	Absolute value for scalars, Cardinality for set
$\lVert\cdot\rVert_{\mathrm{F}}$	Frobenius norm for matrix
$\lVert\cdot\rVert_p$	L_p norm of vector (subscript can be omitted for L_2)

Special Characters

s_i	Line segment (subscript refers to its index in the configuration)
ℓ_i	Length of the line segment
θ_i	Orientation of the line segment
τ_i	Thickness of the line segment
$\mathbf{s}$	Set of line segments (line configuration)
$\#(\mathbf{s})$	Number of line segments in the configuration $\mathbf{s}$
$\mathbf{s}\backslash\{s'\}$	All elements of the set $\mathbf{s}$ except s'

Image Processing

I	Image
$\mathbf{x}_i$	i-th pixel's coordinate $\mathbf{x}_i = (x_i, y_i) \in \mathbb{R}^2$
$\mathbf{G}$	Isotropic Gaussian kernel (matrix) defined in $\mathbb{R}^2$
σ	Smoothing coefficient for Gaussian kernel
γ	Elongation coefficient for antisotropic Gaussian kernel
$\mathbf{F}_\theta$	Steerable filter oriented θ
$*$	Image convolution operator
$\nabla I(\mathbf{x}_i; \sigma^2)$	Image gradient vector at pixel $\mathbf{x}_i$ (σ^2 is for computing derivatives)
$\mathcal{H}(\mathbf{x}_i; \sigma^2)$	Hessian matrix at pixel $\mathbf{x}_i$ (σ^2 is for computing derivatives)

Stochastic Process

$\mathbf{K}(X, X')$	Transition kernel which updates state from X to X'
$\mathbf{K}_{\mathrm{m}}(X, X')$	Sub-transition kernel to support m-type jump from X to X'
q_{m}	Probability to select $\mathbf{K}_{\mathrm{m}}(\cdot, \cdot)$
$\mathbf{Q}_{\mathrm{m}}(X'\lvert X)$	Proposal probability to support m-type jump from X to X'

$\alpha_{\mathrm{m}}(X, X')$	Acceptance ratio to support m-type jump from X to X'

Graph Theory

G	Undirected graph
V	Vertex set, $V = \{v\}$
E	Edge set, $E = \{e_{ij} = (v_i, v_j)\}$
$\|V\|$	The number of vertices in V
$dist(u, v)$	The minimum length of path from u to v in the graph G
$ecc(v)$	Eccentricity of vertex v (*i.e.,* $ecc(v) = \max_{u \in V} dist(u, v)$)
$diam(G)$	Diameter of graph G (*i.e.,* $diam(G) = \max_{v \in V} ecc(v)$)

Machine Learning

$h(\cdot)$	Hypothesis, decision function
$\mathcal{D}$	Training dataset for supervised learning
$\phi(\cdot)$	Feature generating function
$\Psi(\cdot, \cdot)$	Joint feature generating function
z_i	Latent variable
$\hat{z}$	Estimated value obtained via hypothesis
$\Delta(z, \hat{z})$	Loss function to evaluate prediction
ξ	Slack variable

List of Figures

List of Algorithms

List of Tables

Introduction

Introduction

FIELD of research and industry related to computer vision has been developed rapidly recent years due to the growth of computational power and image capturing devices. Now various types of image data are available for analysis. From microscopic images to satellite data, we observe curvilinear structures as different forms. It is required to detect the latent curvilinear structures to improve visibility, to infer biological information, and to extract geographical information. While human can intuitively distinguish these latent curvilinear structures, a technical definition of the curvilinear structure is unclear to date. The difficulties arise because the topology of the curvilinear structures exhibits varied shapes; thus, it is impossible to describe complex curvilinear structure with a few parameters. Also, depending on the acquisition environment, the curvilinear structures can be hidden under the background textures with a low contrast. Therefore, many data representation and computer vision algorithms to detect curvilinear structures have been developed for specific applications. Figure 1.1 shows some curvilinear structures for different applications.

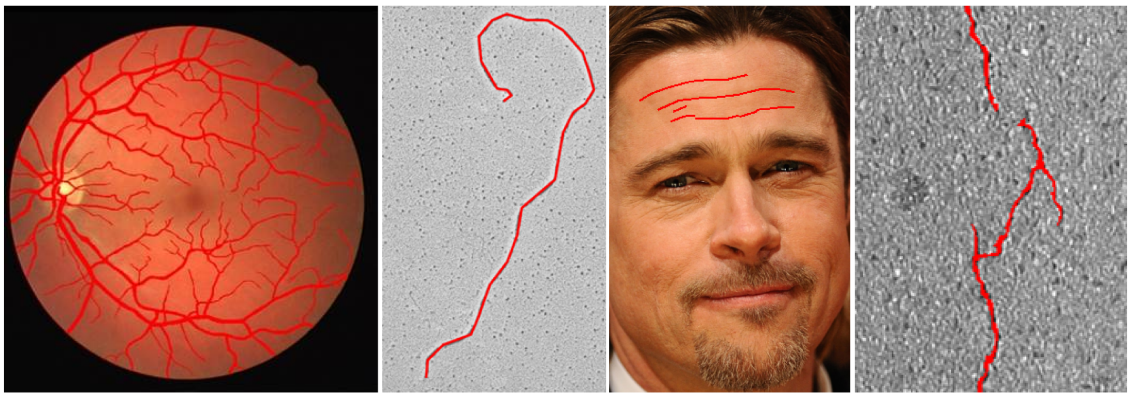


Figure 1.1: Curvilinear structures are observed in different topological features under the various background textures. Left to right images show the latent curvilinear structure in blood vessels, DNA filaments, facial wrinkles, and road cracks.

In early vision, researchers [Freeman 91, Perona 95, Jacob 04] have developed image

convolution filters to decompose curvilinear features from background textures. The main idea behind such filter design is to create line shape templates to extract locally oriented gradient information. Frangi *et al.* analyzes the eigenvalues of the Hessian matrix for a given image to obtain principle directions of the local structure [Frangi 98]. Optimally Oriented Flux (OOF) measures the amount of outgoing gradient flux to find the curvilinear structures [Law 08]. Morphological operator collects pixels according to the structural similarity on the elongated path [Talbot 07]. However, due to the lack of shape interpretation, such methods based on local image features are insufficient to reconstruct underlying curvilinear structure. On the other hand, graphical models such as [González 10, Türetken 13b] define geometric constraints in a local configuration and globally minimize their cost function. More precisely, the graph-based algorithms initialize some points highly corresponding to the latent curvilinear structure, and then define a path which connects these points with geometric priors to provide plausible shapes. Geometric properties of line network are involved as constraint terms when an energy optimization problem is formulated. Stochastic models [Lacoste 05, Jeong 15a] reconstruct curvilinear structures by sampling multiple line segments to maximize a posterior probability of given image data. Similarly to the graph-based representation, geometric priors are considered to define the connectivity and curvature of line segments. Recently, machine learning algorithms have been proposed to detect curvilinear structure. Becker *et al.* [Becker 13] applied a boosting algorithm to obtain an optimal set of convolution filter banks. Sironi *et al.* [Sironi 14] developed a regression model to estimate the scale (width) of the curvilinear structures and to localize the centerlines.

Although the contour grouping algorithms [Tu 06, Arbeáez 11] examine image features corresponding to curves and lines, the goal is quite different from the curvilinear structure reconstruction techniques. The contour grouping algorithms seek closed contour lines to divide an image into meaningful regions. Therefore, the cost function exploits global texture cues in that the contours are associated with salient edges around object boundaries. On the other hand, we look for multiple curvilinear structures, which are not necessarily closed, are latent in the homogeneous texture. Compared with the contour, the curvilinear structures are estimated by subtle local image features. Internal similarity of the structure and an accurate design of shape prior are essential to solve our problem.

1.1 Contributions and outline

In this dissertation we have an ambitious goal to propose a general framework which can analyze varied types of curvilinear structures. We assume that the curvilinear structure can be decomposed into a set of line segments. We hence look for an optimal set

of line segments which corresponds to the latent curvilinear structure. We mainly proposed two curvilinear structure reconstruction algorithms based on stochastic model with sampling technique (Chapter 3) and ranking learning system with graph representation (Chapter 4). For the stochastic model, we propose a posterior probability of line segments for a given image, and optimize the proposed distribution via Markov chain Monte Carlo (MCMC) sampling. Our work specially contributes to reduce hyperparameter dependency for stochastic process. For this purpose, we simultaneous simulate multiple Markov chains with different parameter settings and aggregate multiple line hypotheses. Moreover, we learn a ranking function which measures the comparability of the given line segments and the latent curvilinear structures. A novel graph representation method using the output rankings of the line segments is proposed to show the topological importance of the underlying curvilinear structure. We test various types of datasets, and provide experiment results in each chapter.

1.1.1 Chapter 2: Curvilinear Structure Modeling Reviews

In Chapter 2, we provide the curvilinear feature descriptors based on image gradient and morphological profiles. We perceive the latent curvilinear structure based on inconsistency of background textures and its geometric characteristics. In other words, a sequence of pixels corresponding to the curvilinear structure has different intensity values comparing to its surroundings, and shows thin and elongated shape. Image gradient is computed with respect to various orientations to detect locally oriented features. Moreover, we employ morphological profiles to find pixels showing dissimilar textures to the backgrounds on an elongated path. In this chapter we also review the previous works on curvilinear structure reconstruction models such as segmentation, graph-based representation, stochastic modeling, and machine learning based approaches.

1.1.2 Chapter 3: Stochastic Model

In Chapter 3, we study a stochastic model for curvilinear structure reconstruction. We propose a probability density of line segments for given image, where this probability density consists of data likelihood and prior term. For data likelihood, we exploit rotated image gradient and morphological filtering responses to estimate the presence of curvilinear structure. We approximate the statistics of curvilinear features using Gaussian distribution with support vector machine. We also constrain local geometric behaviors of line segments. We encourage the smooth connection of line segments, whereas we reject the congestion of line segments. To optimize the proposed density, we employ Reversible jump Markov chain Monte Carlo sampler with delayed rejec-

tion. Main contribution of this chapter is to overcome the performance dependency of hyperparameter selection using stochastic optimization. Multiple Markov chains are simulated with different hyperparameter vectors to obtain different line hypotheses, and then we aggregate these hypotheses to reduce sampling space. We provide experiments to compare the curvilinear structure reconstruction results obtained by the proposed algorithm and competing algorithms.

1.1.3 Chapter 4: Inference of Curvilinear Structure

Chapter 4 discusses the structure inference model which quantifies a score according to input-and-output pair information. For classification, the previous works globally set a threshold to remove pixels which show a low probability to be the curvilinear structure. During this process, most of previous works implicitly ignore the correlated information of the pixels on curvilinear structure. We aim to learn a ranking function of the line segments for the given curvilinear structures. The output ranking is able to arrange the line segments in the order of correspondence to the latent curvilinear structures. Specifically, Structured Support Vector Machine (SSVM) is employed to infer the output rankings. We then visualize the topological orders of the curvilinear structure. We study graph theory to first reconstruct curvilinear path which connects the remotest pixels on the latent curvilinear structure. The proposed algorithm progressively adds fine branches into the main structure until it finds the minimum length of structure. We demonstrate experimental results on various datasets to show the efficiency of the proposed algorithm.

1.1.4 Conclusions and Perspectives

We summarize the work and contributions of this thesis. We also provide future directions of the research.

1.1.5 Appendix A: Datasets

Datasets to test the proposed algorithm and competing methods have been included in the appendix.

1.1.6 Appendix B: Thesis overview in French

For French readers, we provide thesis overview of each chapter in French.

1.2 List of Publications

Some of the works described in this dissertation has been published in conference proceedings and submitted to an international journal (IJCV, under review). Please refer to the following list of publications.

Submitted Journal

- S.-G. Jeong, Y. Tarabalka, N. Nisse and J. Zerubia, Inference of curvilinear structure based on learning a ranking function and graph theory, *IJCV* (under review).

Peer-Reviewed Conference Proceedings

- S.-G. Jeong, Y. Tarabalka and J. Zerubia, Stochastic modeling for curvilinear structure reconstruction using morphological profiles, in *Proc. IEEE ICIP*, September 2015.

- S.-G. Jeong, Y. Tarabalka and J. Zerubia, Marked point process model for curvilinear structures extraction, in *EMMCVPR* 2015, LNCS 8932, pp. 436–449, January 2015.

- S.-G. Jeong, Y. Tarabalka and J. Zerubia, Marked point process model for facial wrinkle detection, in *Proc. IEEE ICIP*, pp. 1391–1394, October 2014.

Paper in Preparation

- S.-G. Jeong, Y. Tarabalka, N. Nisse and J. Zerubia, Progressive curvilinear structure reconstruction with structured ranking learning and graph algorithm, in Proc. *IEEE CVPR*, June 2016 (in preparation).

Other Publications

- S.-G. Jeong, C. Lee and C.-S. Kim, Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization, *IEEE TIP*, 22(11):4497–4509, 2013.

- J.-S. Kim, S.-G. Jeong, Y. Joo, and C.-S. Kim, Content-aware image and video resizing based on frequency domain analysis, *IEEE TCE*, 57(2):615–622, 2011.

- S.-G. Jeong, C. Lee and C.-S. Kim, Exemplar-based frame rate up-conversion with congruent segmentation, in *Proc. IEEE ICIP*, pp. 845–848, September 2012.
- S.-G. Jeong and C.-S. Kim, Feature-preserving thumbnail generation based on graph cuts, in *Proc. IEEE ICIP*, pp. 1081–1084, September 2011.

Chapter2

Chapter 2

Curvilinear Structure Modeling: Reviews

Abstract

Many computer vision algorithms have been proposed to analyze underlying curvilinear structures (also called line networks) of image data. This chapter presents the implementation of curvilinear feature descriptors and previous works on curvilinear structure reconstruction models. We summarize the implementation of curvilinear feature descriptors based on image gradient and morphological profiles. Image gradient information is used to extract local image features which are comparable to elongated tubular structures and to estimate orientation. A set of convolution filters is proposed to capture such geometric properties. We perform the morphological operation to find remote pixels on the same curvilinear structure within a pre-defined length. Due to the complex geometric characteristics in different applications, previous works on the curvilinear structure reconstruction modeling have been developed for specific applications. For example, blood vessel networks consist of varied widths and irregular shaped tubular structures. On the other hand, road networks are composed of a constant width and organized structures. Thus, different representation methods have been favored to represent the latent curvilinear structures. This chapter reviews the existing curvilinear structure reconstruction methods with respect to segmentation, graph-based representation, stochastic modeling, and machine learning based approaches.

Contents

THE existing curvilinear structure reconstruction models are employed in many different areas, such as:

- Medical applications for vascular network analysis;
- Biological image analysis;
- Facial wrinkle detection;
- Satellite image analysis for road network extraction; and
- Detection of the defects in the asphalts.

Blood vessel extraction algorithms have been proposed to aid diagnosis of vascular diseases in retinal images [Chaudhuri 89, Frangi 98, Zana 01, Jiang 03, Staal 04, Al-Diri 09, Rigamonti 12]. Retinal images are scanned by ophthalmoscope, which is an optical microscopic device. As the brightness of the scanned images varies according to the illumination environment, image processing techniques are employed to enhance the visibility of blood vessels on the retinal images. The curvilinear structures of vascular images are mainly composed of diameter-varying vessels with irregular shapes. Curvilinear structure segmentation algorithms exploit low-level image features and multiscale manners to analyze vascular networks [Frangi 98, Law 08].

Moreover, curvilinear objects, such as dendrites of neurons and RecA-DNA filaments, are observed in biological images. Neuroimage scientists have attempted to analyze such curvilinear structures in microscopic images [Brown 11]. Specifically, graph-based methods have been proposed to represent the curvilinear structure using tree-like structure [Fischler 81, González 10, Türetken 11, Zhao 11, Wang 11, Peng 11, Türetken 13b]. These algorithms first detect a set of seed points corresponding to the local maxima of tubularity measure of the given image. Then, the seed points are connected to reconstruct underlying curvilinear structures. Surplus paths are corrected by pruning wrong edges which induce topological errors [Peng 11].

Facial wrinkles are an important feature to understand aging progress. In computer vision, facial wrinkles have been studied for age estimation [Fu 10], aged face synthesis [Suo 07, Fu 10], skin texture modeling [Cula 05] and forensic application [Batool 13]. Facial wrinkle detection algorithms could be involved with the above mentioned applications to analyze aged faces in depth. Localization of facial wrinkles is challenging due to complex patterns of wrinkles and skin textures. In [Batool 12, Jeong 14], authors developed stochastic image models which reconstruct the curvilinear structure (facial wrinkles) with a set of disjoint line segments. They employ the image gradient magnitude to extract linearity feature and constrain the spatial distribution of the line segments.

In remote sensing, road network extraction algorithms have been explored in order to interpret the geographical information [Vosselman 95, Geman 96, Lacoste 05, Hu 07, Valero 10]. Based on the imaging sensor types equipped on a satellite, images have different spatial and spectral resolutions. Road extraction algorithms employ prior knowledge to describe geometric characteristics of the road networks. Geometrically, roads are elongated and have a limited curvature [Vosselman 95, Hu 07]. Also, width of the structure is generally constant because the road networks are man-made systems. Automation is an important issue for road extraction algorithms in that the remote sensing image covers wide urban areas. Early developed road networks tracing algorithms require a manual drawing of road networks for an initial condition [Vosselman 95, Geman 96]. Lacoste *et al.* [Lacoste 05] developed a fully automatic algorithm based on marked point process framework [Lieshout 00, Descombes 02, Chatelain 12].

Defects on road surfaces (road cracks) appear on the thin lines heading in similar direction. For the purpose of convenience and safety, it is preferred to evaluate road surface conditions based on image processing techniques. The task to detect road crack on the images is difficult due to the optical noises and the rough surface textures. Morphological operator to detect cracks is studied with respect to its spatial patterns [Iyer 05]. Chambon *et al.* formulated an image segmentation problem following the Markov Random Field (MRF) property to detect irregular shape of the road crack [Chambon 10]. In addition, the acquisition systems which combine optical sensor and laser scanner have been developed to measure the quality of road surfaces. An intensive review of both acquisition systems and image processing techniques for road cracks evaluation can be found in [Chambon 11],

In the following sections, we first explain the way to compute curvilinear feature descriptors based on image gradient and image morphological profiles (Section 2.1). We then review the existing curvilinear structure reconstruction methods which are segmentation, graph-based model, stochastic model, and machine learning approaches (Section 2.2).

2.1 Curvilinear Feature Descriptors

An image $I : \mathbb{R}^2 \mapsto \mathbb{R}$ can be regarded as a function which maps a pixel coordinate into an intensity value. Suppose that an image contains a curvilinear structure of an arbitrary shape. In the previous works, image gradient and morphological features have been employed to describe curvilinear structures which are latent in natural images. The geometric characteristics of a curvilinear structure have important roles and must be taken into account when developing a curvilinear feature descriptor. Geometrically speaking, curvilinear structures are rotatable and have an elongated tube-like shape.

Image gradient information obtained by convolution operation is useful to estimate local orientation and to describe the shape of the curvilinear structure. Also, the curvilinear structures are consecutively distributed toward a similar direction. Since morphological operation is able to recursively detect structural elements of the image, it highlights a sequence of pixels on the underlying curvilinear structure. In the remainder of this section, we review how to implement curvilinear feature descriptors based on image gradient and morphological profiles in detail.

2.1.1 Image gradient profiles

Convolution filtering is a basic operation in image processing to extract low-level image features. For example, Gabor filter banks are popularly employed to detect object boundaries [Mehrotra 92] and to classify fine-grained textures [Jain 90, Grigorescu 02]. Intensive studies to analyze scale space analysis for edge and ridge detection have been done by [Lindeberg 98]. Specifically, researchers have designed rotatable filters (also known as steerable filters) to estimate local orientation of curvilinear structures [Freeman 91, Perona 95, Jacob 04]. An efficient design of rotatable filters is proposed to reconstruct an arbitrarily oriented kernel as a linear combination of basis kernels. Thus, we perform a set of convolution filters to analyze the given image in different orientations (see Figure 2.1).

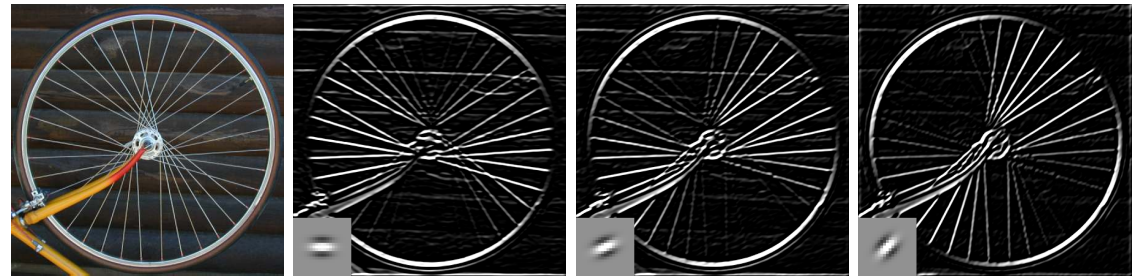
The main idea behind such convolution filter design is to create line shaped templates which are analogous to the underlying curvilinear structures. Second order derivative of anisotropic Gaussian kernel is a good approximation to describe shape of the tubular structures. Moreover, the orientation of the anisotropic Gaussian kernel is easily controllable; this property is helpful to estimate the local orientation of the curvilinear structure.

We first analytically develop convolution filter banks to extract curvilinear features. Let $\mathbf{G} : [-\frac{W}{2}, \frac{W}{2}]^2 \mapsto \mathbb{R}$ be an isotropic Gaussian kernel:

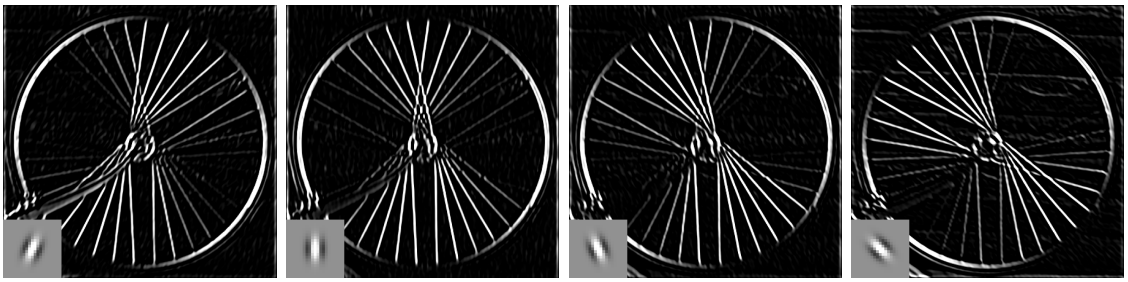
$$\mathbf{G}_{\sigma^2} : \mathbf{G}(x, y; \sigma^2) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right), \tag{2.1}$$

where x and y denote the distance from origin $(0, 0)$ in horizontal and vertical axis, respectively. W denotes size of the kernel, and σ denotes the smoothing factor of the Gaussian kernel. Adding an elongation parameter $\gamma > 1$ makes the isotropic Gaussian become anisotropic for x axis:

$$\mathbf{G}_{\sigma^2,\gamma} : \mathbf{G}(x, y; \sigma^2, \gamma) = \exp\left(-\frac{x^2 + (y/\gamma)^2}{\sigma^2}\right). \tag{2.2}$$



(a) Input, I (b) $\mathbf{F}_{\theta_1} * I$, $\theta_1 = 0°$ (c) $\mathbf{F}_{\theta_2} * I$, $\theta_2 = 22.5°$ (d) $\mathbf{F}_{\theta_3} * I$, $\theta_3 = 45°$



(e) $\mathbf{F}_{\theta_4} * I$, $\theta_4 = 67.5°$ (f) $\mathbf{F}_{\theta_5} * I$, $\theta_5 = 90°$ (g) $\mathbf{F}_{\theta_6} * I$, $\theta_6 = 112.5°$ (h) $\mathbf{F}_{\theta_7} * I$, $\theta_7 = 135°$

Figure 2.1: The local orientation of the given image I is analyzed by performing image convolution with rotatable filter banks. Filtering responses accentuate pixels according to the specified orientation. The associated convolution filters $\mathbf{F}_\theta$ are depicted in left bottom of each image.

The local orientation is taken into account by algebraically performing a rotation in $\mathbb{R}^2$. That is $x_\theta = x\cos(\theta) - y\sin(\theta)$ and $y_\theta = x\sin(\theta) + y\cos(\theta)$:

$$\mathbf{G}_{\sigma^2,\gamma,\theta} : \mathbf{G}(x, y; \sigma^2, \gamma, \theta) = \exp\left(-\frac{x_\theta^2 + (y_\theta/\gamma)^2}{\sigma^2}\right). \tag{2.3}$$

Finally, partial derivative of (2.3) with respect to x_θ is performed to build a candidate convolution filter $\mathbf{F}_\theta : [-\frac{W}{2}, \frac{W}{2}]^2 \mapsto \mathbb{R}$ that accentuates image gradient magnitudes for θ direction:

$$\mathbf{F}_\theta = \frac{\partial^2}{\partial x_\theta^2}\mathbf{G}_{\sigma^2,\gamma,\theta} \tag{2.4}$$

$$\frac{\partial^2}{\partial x_\theta^2}\mathbf{G}(x, y; \sigma^2, \gamma, \theta) = \left(\frac{4x_\theta^2}{\sigma^4} - \frac{2}{\sigma^2}\right)\exp\left(-\frac{x_\theta^2 + (y_\theta/\gamma)^2}{\sigma^2}\right). \tag{2.5}$$

Alternatively, we show a constructive method to build a rotatable convolution filter banks as the combination of shifted anisotropic Gaussian kernels. An anisotropic Gaussian kernel is shifted to the aperture size a with respect to the minor axis direction

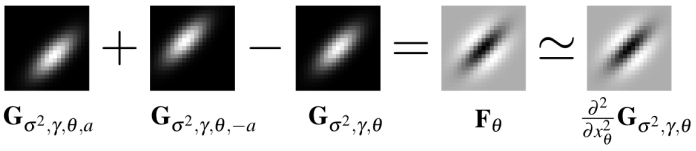


Figure 2.2: A candidate kernel is generated by the linear combination of shifted anisotropic Gaussian kernels. This approach is employed to analyze the size of underlying curvilinear structure intuitively.

as follows:

$$\mathbf{G}_{\sigma^2,\gamma,\theta,a} : \mathbf{G}(x, y; \sigma^2, \gamma, \theta, a) = \exp\left(-\frac{(x_\theta - a)^2 + (y_\theta/\gamma)^2}{\sigma^2}\right). \tag{2.6}$$

Recall that a candidate filter $\mathbf{F}_\theta$ is Mexican-hat like shape to extract tubular structures. We combine shifted anisotropic Gaussian kernels to imitate the second derivative operation such as:

$$\mathbf{F}_\theta = \mathbf{G}_{\sigma^2,\gamma,\theta,a} + \mathbf{G}_{\sigma^2,\gamma,\theta,-a} - \mathbf{G}_{\sigma^2,\gamma,\theta}. \tag{2.7}$$

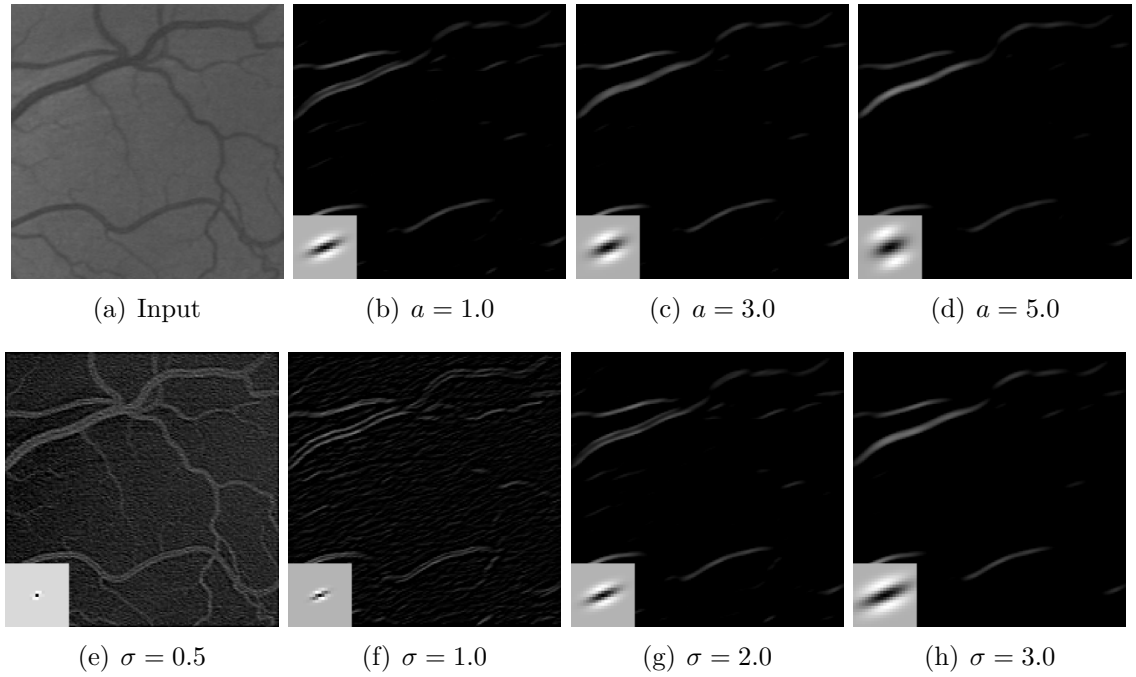
Figure 2.2 visually compares a candidate filter created by constructive approach and analytic approach.

Comparing these two approaches to create convolution filter banks, the constructive approach (2.7) is apt to estimate the size of curvilinear structures. While the smoothing factor σ of Gaussian kernel is related to the size of the curvilinear structure, it is more likely to blur image signals. On the other hand, the large aperture value is used to highlight pixels on wide curvilinear structure and a small value corresponds to thin structures (see Figure 2.3).

2.1.2 Image morphological profiles

Because image convolution is defined to exploit spatial neighboring pixels, the output is apt to detect local geometric features of the curvilinear structures. However, in the case of uneven illuminations and heavy noises, local image features fail to classify pixels on the same curvilinear structure if they are located far from each other. To make up the disadvantages using local image features to describe curvilinear structure, we compute a morphological operator to exclude pixels corresponding to non-curvilinear structures. To detect thin and elongated shape structure, morphological operations have been designed for medical [Zana 01], remote sensing [Valero 10] and biological [Zhao 11] applications.

The morphological operation is performed to manipulate binary images accord-



(a) Input (b) $a = 1.0$ (c) $a = 3.0$ (d) $a = 5.0$

(e) $\sigma = 0.5$ (f) $\sigma = 1.0$ (g) $\sigma = 2.0$ (h) $\sigma = 3.0$

Figure 2.3: We compare convolution filter banks which are built constructively (**top row**) and analytically (**bottom row**). Increasing smoothing factor σ is related to estimating width of the structure, it is more likely to remove noises. On the other hand, controlling the aperture value a of constructive filters provides consistent result that highlight pixels on the curvilinear structure according to the width. The associated convolution filters are depicted in left bottom of each image.

ing to the structural similarity. Recall that *opening*, a basic morphological operation, removes objects, which are brighter than their surrounding and smaller than a structuring element. Similarly, *path opening* operator [Talbot 07] excludes pixels which do not belong to the curvilinear structures. Let represent an image as a graph, where each pixel corresponds to a node on the graph. The structural elements of curvilinear structure is defined on adjacency graphs with the consideration of topological characteristics, *i.e.,* thin and elongated path. Figure 2.4 (a) shows an adjacency graph consisting of west-east cones (W-E graph) which is used to find horizontal paths. To detect vertical paths, W-E graph is rotated in 90° and then path opening is applied.

For a given binary image $B : \mathbb{R}^2 \mapsto \{0, 1\}$, path opening seeks for valid paths on the adjacency graph. Let $\rho_L(u) = \{v \in B, u \to v\}$ be the structuring function which defines a path with a given length L, where $u \to v$ denotes a directed edge going from node u to v on the adjacency graph. Then, a valid ρ_L-path of length L is represented as L-tuple $\mathbf{u} = (u_1, \ldots, u_L)$ that satisfies $u_{k+1} \in \rho_L(u_k)$ for $k = 1, \ldots, L-1$. The path

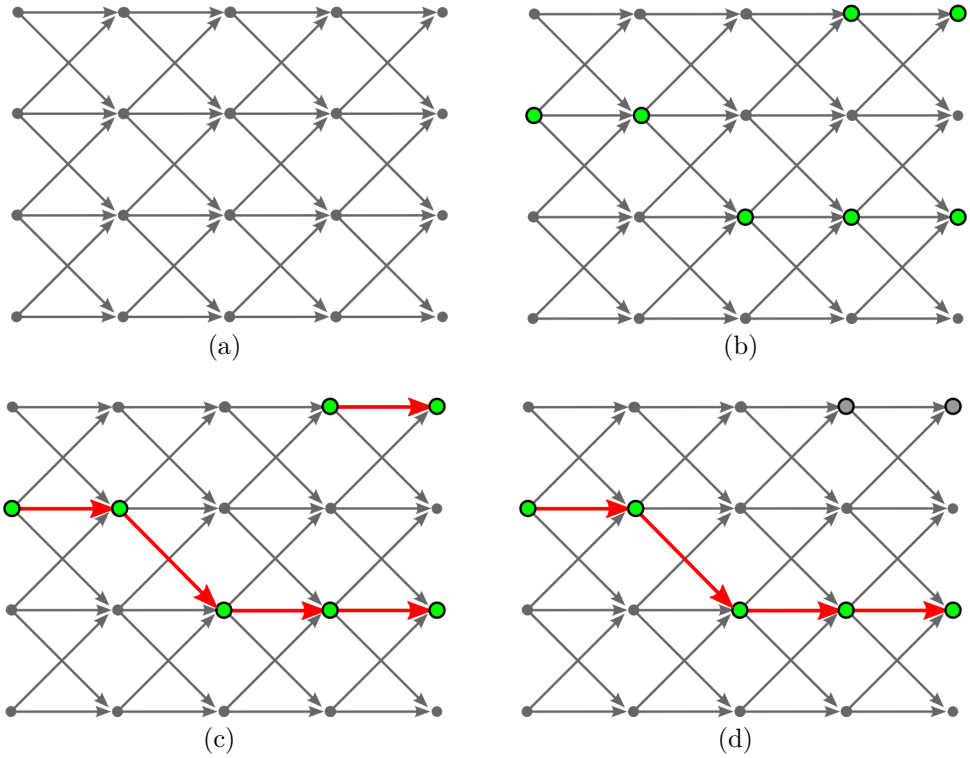
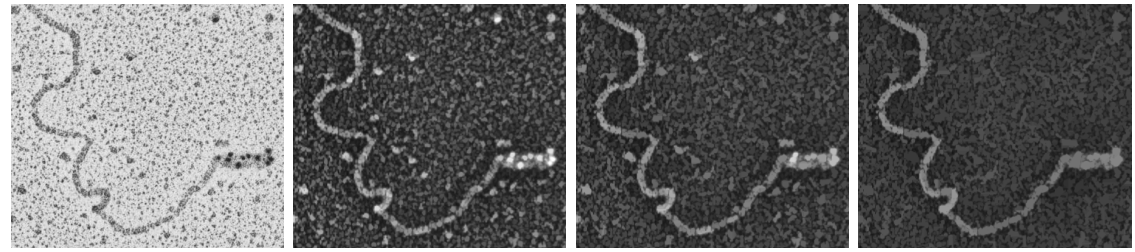


Figure 2.4: Toy example to show how the path opening operator finds valid paths on the adjacency graph: (a) an adcaency graph composed with south-north cones, (b) greed nodes are associated with pixels after image binarization, (c) seeking paths, denoted red color, on the graph, and (d) a path is valid if its length is larger than pre-defined length L.

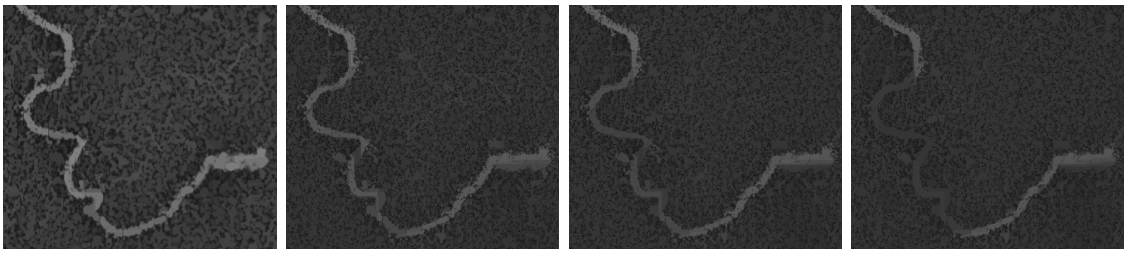
opening operator is defined as the union of all valid ρ_L-paths of L in the binary image.

To generalize the binary path opening operator to the case of gray-scale images, we decompose an input gray level image into binary images with decreasing order of threshold values. For each decomposed binary image, the binary path opening operator is performed to find the valid paths. The infimum threshold value employed to create the binary image becomes the output of gray level morphological filtering. Finally, morphological profiles are composed of path opening results with different length parameters L. Figure 2.5 shows morphological filtering results on a DNA image.

The curvilinear structure descriptors are employed to measure the quality of reconstructed curvilinear structures based on the image data. However, it is ambiguous to discern pixels corresponding to the curvilinear structure from object boundaries or high frequency noise components. To improve performances, geometric prior and machine learning techniques are involved to advanced curvilinear structure reconstruction



(a) Input, I (b) $L = 10$ (c) $L = 20$ (d) $L = 40$



(e) $L = 60$ (f) $L = 120$ (g) $L = 180$ (h) $L = 240$

Figure 2.5: **Morphological profiles**: Path opening [Talbot 07] results with different length parameter L. As the length parameter is increased, the path opening operator removes small objects which are not on a valid path.

modeling. In the following section, we review the state-of-the-art works on curvilinear structure reconstruction modeling.

2.2 Existing Curvilinear Structure Reconstruction Methods

To compute a likelihood probability whether a pixel is on the latent curvilinear structure, segmentation techniques can be applied. Low-level image features,*e.g.,* image gradients, are mainly used to estimate this likelihood. On the other hand, graph-based models interpret the curvilinear structure as a sequence of nodes on a graph. They exploit low-level curvilinear features to localize nodes on the curvilinear structure, and define geometric constraints to reconstruct the curvilinear structure with plausible shape. Bottom-up approach represents the curvilinear structure as a disjoint set of line segments. For this purpose, stochastic models have been proposed to define the underlying distribution of the line segments, where an optimal solution consists of smoothly connected line segments. Furthermore, machine learning based approaches are able to detect curvilinear structures on various datasets. We have grouped the existing curvilinear structure reconstruction methods into four classes, which are reviewed

in the following sections:

1. Curvilinear structure segmentation [Frangi 98, Law 08]
2. Graph-based models [González 10, Türetken 13b]
3. Stochastic image models [Lacoste 05]
4. Machine learning based approaches [Becker 13, Sironi 14]

2.2.1 Curvilinear structure segmentation

Curvilinear structure segmentation algorithms can be used to describe irregular shape of the curvilinear structure by evaluating a local tubularity score $\mathsf{S}(\mathbf{x}_i)$ at pixel $\mathbf{x}_i$, *i.e.,* a score corresponds to the likelihood for their pixel to belong to a curvilinear structure. Then, a threshold value is set to decompose the curvilinear structure. To consider varying widths of the curvilinear structure, researchers [Frangi 98, Law 08, Türetken 13a] evaluated their tubularity scores over different scale (width) parameters. The multiscale version tubularity scores $\mathsf{S}^{\mathrm{MS}}(\mathbf{x}_i)$ are maximized at a scale factor r corresponding to the underlying width of the structure at each pixel:

$$\mathsf{S}^{\mathrm{MS}}(\mathbf{x}_i) = \max_{r_{\min} \leq r \leq r_{\max}} \mathsf{S}(\mathbf{x}_i; r). \tag{2.8}$$

We briefly review popular curvilinear structure segmentation algorithms which are mainly applied to vascular network extraction: Hessian matrix analysis [Frangi 98] and Optimally Oriented Flux [Law 08].

Hessian matrix analysis

In many engineering problems, Taylor approximation is widely employed to analyze the given function as the sum of an infinite series. To interpret the local image structures, Taylor expansion of the image function is given by:

$$I(\mathbf{x}_i + \delta\mathbf{x}_i; r) \simeq I(\mathbf{x}_i; r) + \delta\mathbf{x}_i^{\intercal} \nabla I(\mathbf{x}_i; r) + \delta\mathbf{x}_i^{\intercal} \mathcal{H}(\mathbf{x}_i; r) \delta\mathbf{x}_i, \tag{2.9}$$

where $\mathbf{x}_i = (x_i, y_i)$ denotes the i-th pixel of the given image I, $\delta\mathbf{x}_i$ is an infinitesimal distance from $\mathbf{x}_i$, r is a scale factor, and $\nabla I(\mathbf{x}_i; r)$ is the gradient vector of $I(\mathbf{x}_i; r)$. Note that image gradient is computed by convolution with derivative of Gaussian kernel. That is

$$\nabla I(\mathbf{x}_i; r) = \begin{bmatrix} \frac{\partial}{\partial x} I(\mathbf{x}_i; r) \\ \frac{\partial}{\partial y} I(\mathbf{x}_i; r) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} \left(\mathbf{G}_{|\sigma^2 = r} * I \right)(\mathbf{x}_i) \\ \frac{\partial}{\partial y} \left(\mathbf{G}_{|\sigma^2 = r} * I \right)(\mathbf{x}_i) \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial}{\partial x} \mathbf{G}_{|\sigma^2 = r} \right) * I(\mathbf{x}_i) \\ \left(\frac{\partial}{\partial y} \mathbf{G}_{|\sigma^2 = r} \right) * I(\mathbf{x}_i) \end{bmatrix}. \tag{2.10}$$

The Hessian matrix $\mathcal{H}(\mathbf{x}_i; r)$ is related to the second order derivative of the image, which is computed by

$$\mathcal{H}(\mathbf{x}_i; r) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} I(\mathbf{x}_i; r) & \frac{\partial^2}{\partial x \partial y} I(\mathbf{x}_i; r) \\ \frac{\partial^2}{\partial x \partial y} I(\mathbf{x}_i; r) & \frac{\partial^2}{\partial y^2} I(\mathbf{x}_i; r) \end{bmatrix}. \tag{2.11}$$

Intuitively, the second derivative of the image (Hessian) reflects the local image geometries. Hence, researchers examined the eigenvalues of the Hessian matrix to enhance curvilinear structure in the vascular images [Koller 95, Sato 97, Frangi 98]. More precisely, for 2D images, the eigenvalues of the Hessian matrix, λ_1 and λ_2, show the principle directions of the local structure. With the consideration of the line-shaped structure, pixels on the curvilinear structure are locally aligned toward the same direction. Frangi *et al.* [Frangi 98] proposed a vesselness measure using the geometric ratio of the eigenvalues $R = \frac{\lambda_1}{\lambda_2}$, where $|\lambda_1| \leq |\lambda_2|$. Also, the presence of the curvilinear structure (blood vessel) is obtained by the magnitude of the eigenvalues, which are associated with Frobenius norm of the Hessian matrix, $\|\mathcal{H}\|_F = \sqrt{\sum_{j=1}^{2} \lambda_j^2}$. In [Frangi 98], the likelihood of being the curvilinear structure is proposed by the combination of such two observations:

$$\mathsf{S}^{\mathrm{H}}(\mathbf{x}_i; r) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp\left(-\frac{R(\mathbf{x}_i;r)^2}{2\beta^2}\right)\left(1 - \exp\left(-\frac{\|\mathcal{H}(\mathbf{x}_i;r)\|_F^2}{2c^2}\right)\right) & \text{otherwise,} \end{cases} \tag{2.12}$$

where β and c control the sensitivity of the corresponding measures R and $\|\mathcal{H}\|_F$, respectively. To take into account different scales (widths) of the curvilinear structure, the vesselness measure (eq. 2.16) introduces the scale factor r which is related to the bandwidth of the filters to compute image derivatives. While the approaches to analyze the eigenvalues of the Hessian for evaluating the local tubularity score of pixels are simple, it is hard to generalize them if the image is heavily corrupted by noise. Also, the results often contain ringing artifacts around curvilinear structure boundaries.

Image gradient flux

Other works, such as [Vasilevskiy 01, Bouix 05, Law 08, Türetken 13a], analyze the local image gradient flux for detecting curvilinear structures. For simplicity, it is assumed that pixels on the curvilinear structures are brighter than backgrounds. Then, at the object boundaries, inward gradient flux is minimized (the outgoing flux is maximized in contrary). Recall that the shape of the curvilinear structure is symmetrical for its cross section and the boundaries are in contrast with the background textures. Therefore, the amounts of the image gradient flux within a local spherical region indi-

cate the presence of the curvilinear structure. Meanwhile, the direction of the gradient flux is orthogonal to the local structure.

Law and Chung [Law 08] proposed the Optimally Oriented Flux (OOF) which measures projected image gradient flux along an optimal direction at a local spherical region. In [Schey 05], the outgoing image gradient flux I_f is defined as the surface integral of gradient vectors toward the direction of $\mathbf{x}$ over a surface S with radius r:

$$I_f(\mathbf{x}_i; r, \mathbf{v}) = \int_{\partial S} ((\nabla I(\mathbf{x}_i + r\mathbf{n}) \cdot \mathbf{v})\mathbf{v}) \cdot \mathbf{n} dA = \mathbf{v}^\intercal \mathbf{Q}(\mathbf{x}_i; r)\mathbf{v}, \tag{2.13}$$

where dA is the infinitesimal area on ∂S, and $\mathbf{n} = [n_1 \ n_2]^\intercal$ is the unit normal vector of the surface ∂S. The matrix $\mathbf{Q}(\mathbf{x}_i; r)$ encodes the amount of gradient flux projected onto the surface S with the normal vector $\mathbf{n}$, which is given by

$$\mathbf{Q}(\mathbf{x}_i; r) = \begin{bmatrix} \int_{\partial S} \frac{\partial}{\partial x} I(\mathbf{x}_i + r\mathbf{n}) n_1 dA & \int_{\partial S} \frac{\partial}{\partial x} I(\mathbf{x}_i + r\mathbf{n}) n_2 dA \\ \int_{\partial S} \frac{\partial}{\partial y} I(\mathbf{x}_i + r\mathbf{n}) n_1 dA & \int_{\partial S} \frac{\partial}{\partial y} I(\mathbf{x}_i + r\mathbf{n}) n_2 dA \end{bmatrix}. \tag{2.14}$$

In practice, the matrix $\mathbf{Q}(\mathbf{x}_i; r)$ is implemented with the set of linear filtering responses, where the linear filters are second derivatives of Gaussian kernel along the corresponding axes. The optimal direction of OOF is computed by minimizing the inward flux at the local spherical structure. It is equivalent to the solution of the following constrained optimization problem:

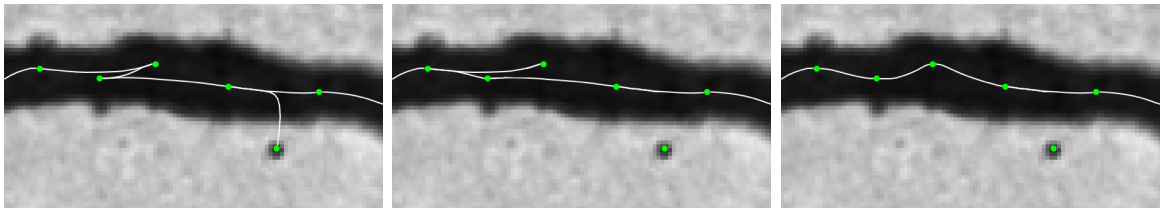
$$\begin{array}{ll} \max\limits_{\mathbf{v}} & \mathbf{v}^\intercal \mathbf{Q}(\mathbf{x}_i; r)\mathbf{v} \\ \text{subject to} & \mathbf{v}^\intercal \mathbf{v} = 1 \end{array}. \tag{2.15}$$

With the Lagrange multiplier method, (2.15) is solved as a generalized eigenvalue problem [Boykov 04]. Finally, Law and Chung [Law 08] employed a geometric mean of the eigenvalues of matrix $\mathbf{Q}(\mathbf{x}_i; r)$ to evaluate tubularity score at $\mathbf{x}_i$ with scale factor r:

$$\mathsf{S}^{\mathrm{OOF}}(\mathbf{x}_i; r) = \begin{cases} \sqrt{|\lambda_1 \lambda_2|} & \lambda_1 \leq \lambda_2 < 0, \\ 0 & \text{otherwise}, \end{cases} \tag{2.16}$$

When compared with the methods based on the Hessian matrix analysis, computation of the OOF is less dependent on convolution operations to compute the second derivative of the image.

The curvilinear structure segmentation algorithms finally provide a likelihood probability (tubularity score) of being on the curvilinear structure for each pixel. The tubularity score is employed to enhance the visibility of curvilinear structure on the image. However, it is ambiguous to decide the probability bound based on the observation of the likelihood. For instance, a low threshold value reduces the precision to represent the curvilinear structure. To resolve this issue, geometric priors and global energy



(a) (b) (c)

Figure 2.6: The graph-based models reconstruct curvilinear structures by connecting seed points (denoted by green points), which are associated with local maxima of tubularity scores. (a) The log-likelihood of the image based term $-\log P(I, G|\mathbf{t})$ is minimized to recontruct path, which shows topological errors (zig-zags and surplus branch). (b) Geometric prior $-\log P(\mathbf{t})$ is incorperated to correct such topological erors. (c) Additional relaxation term is required to prevent counting a seed point twice for path reconstruction. Images courtesy of [Türetken 11].

optimization are included into curvilinear structure reconstruction modeling.

2.2.2 Graph-based models

Topology of the curvilinear structure often refers to a tree-like structure. Graph based models [Fischler 81, González 10, Türetken 11, Zhao 11, Wang 11, Peng 11, Türetken 13b] reconstruct the curvilinear structure with geodesic paths in a graph $G = (V, E)$, where $V = \{v_i\}$ denotes a set of vertices and $E = \{e_{ij} = (v_i, v_j)\}$ is a set of edges. For an efficient computation, the vertex set V (also called seed points) is subsampled from the whole pixels of the given image. Topologically speaking, the seed points are willing to be junctions of the curvilinear structure, and the tubularity scores are locally maximized at these points. In practice, thinning operator [Lee 94] is performed with respect to curvilinear segmentation maps or maximum intensity projection images to detect junction points.

Various methods have been proposed to trace an optimal path that connects seed points on the graph. Sargin2011 *et al.* developed a Hidden Markov Model (HMM) to estimate a curvilinear path [Sargin 11]. Peng *et al.* intially reconstructed an over-complete tree via Dijkstra's algorithm [Dijkstra 59] and discarded redundant nodes [Peng 11]. Global energy optimization problem is formulated to maximize *a posteriori* probability of the image data and geometric constraints [González 10,

Türetken 11, Türetken 13b]:

$$\mathbf{t}^* = \underset{\mathbf{t}\in G}{\operatorname{argmax}}\, p(\mathbf{t}|I, G) \tag{2.17}$$

$$= \underset{\mathbf{t}\in G}{\operatorname{argmax}}\, p(I, G|\mathbf{t})p(\mathbf{t}), \tag{2.18}$$

where $\mathbf{t}$ is a sequence of vertices which are chosen to reconstruct a curvilinear structure. Thus, the first term $p(I, G|\mathbf{t})$ measures the quality of the reconstructed path based on the image data. $p(\mathbf{t})$ is designed to regularize the geometric properties of the tree structure, *e.g.,* connectivity, orientation consistency, and crossover. The solution of (2.18) can be obtained by either Ant Colony Optimization (ACO) [Türetken 11] or integer programming [Türetken 13b]. Although the geometric priors are employed, an additional relaxation term is needed to modify the topological errors (see Figure 2.6).

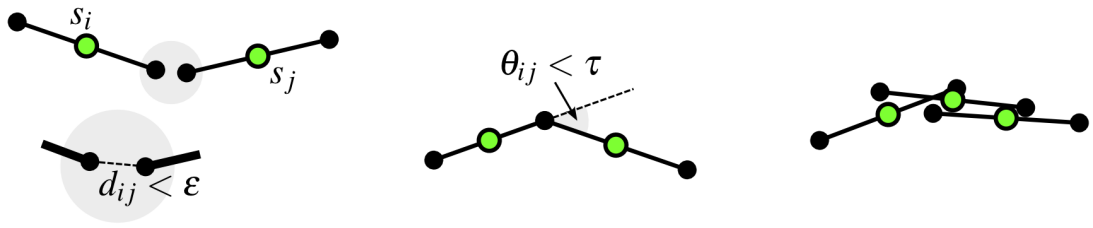
2.2.3 Stochastic image models

Curvilinear structures can be seen as a combination of small line segments. Stochastic models define a distribution of line segments with geometric priors to detect multiple line segments corresponding to the curvilinear structure [Lacoste 05, Tu 06, Schlecht 07, Batool 12, Predoehl 13, Jeong 14]. Each line segment s encodes its shape information, *i.e.,* length and orientation. Thus, a line segment s is defined as a tuple of the center coordinate $\mathbf{x}$, length ℓ, and orientation θ. That is $s = (\mathbf{x}, \ell, \theta) \in I \times \mathbb{R} \times \Theta$, where Θ is set of elements in $[0,\ 2\pi[$. A probability density function of line segments $p(\mathbf{s}|I)$ is proposed to fit line segments into the given image data and to constrain spatial relationship of the line segments. Thus, the probability density function consists of a data dependent term $p(I|\mathbf{s})$ and a prior energy $p(\mathbf{s})$ given by

$$p(\mathbf{s}|I) \propto p(I|\mathbf{s})p(\mathbf{s}), \tag{2.19}$$

where $\mathbf{s} = \{s_1, \ldots, s_n\}$ denotes a configuration of the line segments. More specifically, to localize line segments onto the curvilinear structure, the data term $p(I|\mathbf{s})$ exploits curvilinear features such as local statistics of pixel intensities [Lacoste 05], Gabor filtering responses [Tu 06], and image gradients [Batool 12, Jeong 14]. The prior energy $p(\mathbf{s})$ defines local interaction of the line segments in terms of the connectivity and the curvature (see Figure 2.7).

To optimize the probability density, Reversible Jump Markov chain Monte Carlo sampler [Green 95] has been typically employed. A state of the Markov chain is related to a line configuration. The Markov chain is locally perturbed by sub-transition kernels, and converges at a stationary state which maximizes the probability density (see Chapter 3 for details).



(a) Connectivity (b) Curvature (c) Congestion

Figure 2.7: Stochastic models constain the spatial distribution of the line segments, where each line segment s_i is represented as a tuple of its center coordinate $\mathbf{x}_i$, length ℓ_i, and orientation θ_i. (a) Pair line segments are assumed to be connected if the end-to-end distance d_{ij} of them is relatively small ($< \epsilon$). (b) The angular difference θ_{ij} of two line segments is related to the curvature. A small curvature is preferred in order to reconstruct smoothly connected curvilinear structures. (c) Congestion of line segments will be penalized.

However, the stochastic models require heavy formalization to interpret spatial distribution of the line segments. Large number of parameters should be specified to describe the shape of the line segments and to control the relative importance of data and prior energy terms. Stochastic modeling is considered less practical for applying to a large amount of varied datasets because its performance is very sensitive to the selection of parameters.

2.2.4 Machine learning based approches

Previous works on curvilinear structure modelling have been developed independently to tackle a specific application. Recent trends on computer vision research proceed to handle many data as possible. Thus, it is desired that the solution should be self-contained and use the minimum number of user-defined parameters. Apparently, for this purpose, machine learning techniques are involved with many computer vision problems nowadays. Specifically, model parameters can be learned automatically with labeled dataset (training set), so that we can reduce efforts to adapt the proposed model for different types of dataset. Becker *et al.* [Becker 13] opened the possibility to employ a machine learning technique for curvilinear structure segmentation. They proposed a Gradient boosting framework to obtain optimal convolution filter banks to extract the curvilinear features. Scale space distance transform (SSDT) is proposed as a regression function to predict the scale (width) of the curvilinear structure and localize centerlines [Sironi 14] .

In the general setup of the machine learning problems, we aim to learn a function

$h : \mathcal{X} \mapsto \mathcal{Z}$ between an input space $\mathbf{x} \in \mathcal{X}$ and output space $z \in \mathcal{Z}$. To evaluate the quality of a prediction $\hat{z} = h(\mathbf{x})$, a loss function $\Delta(z, h(\mathbf{x}))$ is defined. Then, a machine learning algorithm minimizes the loss function to find an underlying distribution of data with a given training dataset $\mathcal{D} = \{(\mathbf{x}_i, z_i)\}_{i=1}^{N}$, where N is the number of input and output pairs.

Gradient boosting [Zheng 07, Hastie 09] approximates a function as a linear combination of weak classifiers:

$$h(\mathbf{x}) = \sum_{j=1}^{M} \alpha_j \varphi_j(\mathbf{x}), \tag{2.20}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes an input vector, $\alpha_j \in \mathbb{R}$ is a weight of the weak classifier $\varphi_j(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$, and M is the number of week classifiers. For a given training dataset $\mathcal{D}$, an optimal parameter set of regression tree h is obtained in a greedy manner. In other words, at each weak classifier φ_j, the corresponding weights are found via exhaustive search on a regression tree to minimize a total loss $\mathcal{L} = \sum_{i=1}^{N} \Delta(z_i, h(\mathbf{x}_i))$.

Supervised curvilinear feature learning

In [Becker 13], Gradient boosting framework is employed to create convolution filter candidates with a closed form, and to learn a regression tree for the curvilinear structure segmentation. For the curvlinear feature learning and classification, Gradient boosting implementation of [Becker 13] consists of two phases: (1) searching for a set of convolution kernels and (2) learning parameters of the regression tree which minimize the splitting cost. A weak classifier $\varphi_j(\mathbf{x}_i)$ corresponds to the inner product of a convolution kernel $\hat{\mathbf{F}}_p$ and a vectorized image patch $\mathbf{X}_i$ at a sampled point $\mathbf{x}_i$. Thus, a candidate kernel is found:

$$\hat{\mathbf{F}}_p = \underset{\mathbf{F}}{\operatorname{argmin}} \sum_{i=1}^{N} \omega_i \left(\mathbf{F}^{\intercal}\mathbf{X}_i - g_i\right)^2 + \sum_{m \sim n} \left(\mathbf{F}^{(m)} - \mathbf{F}^{(n)}\right)^2, \tag{2.21}$$

where ω_i and g_i denote weight and target values, respectively. The partial differentiation of the loss function is computed to obtain ω_i and g_i. The second term of eq. (2.21) induces smooth kernel values with local neighborhoods $m \sim n$, where $\mathbf{F}^{(m)}$ denotes m-the value in the filter $\mathbf{F}$. Then, regression parameters for decision tree are computed with respect to each $\hat{\mathbf{F}}_p$.

Centerline detection based on learning a scale-space distance transfrom

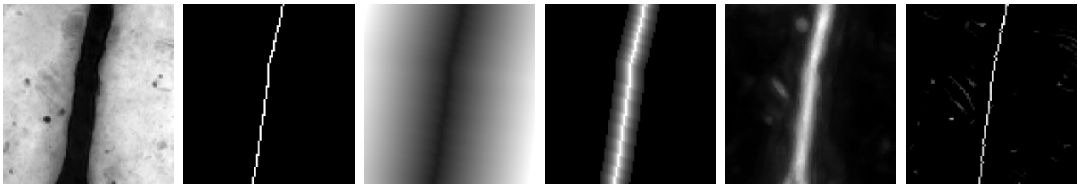
Gradient Boosting framework is exploited in a different way to localize a centerline of the curvilinear structure in [Sironi 14]. In this work, the centerline also encodes the scale of the curvilinear structures. Previous curvilinear structure segmentation methods

often highlight pixels more on the boundaries of the structures than on the centerlines, even though the models take into account multiscale terms. Thus, in [Sironi 14], the authors proposed a regression model which learns the scale of the curvilinear structure based on the distance transform.

Let $D_C(\mathbf{x}_i)$ denote the distance transform which measures the distance between the given point $\mathbf{x}_i$ and the closest point belonging to the centerline. To train a regressor, the distance transform $D_C(\mathbf{x}_i)$ is updated into a function $d(\mathbf{x}_i)$, whose values monotonically decrease from the centerline:

$$d(\mathbf{x}_i) = \begin{cases} \exp\left\{6\left(1 - \frac{D_C(\mathbf{x}_i)}{d_M}\right)\right\} - 1, & \text{if } D_C(\mathbf{x}_i) < d_M \\ 0 & \text{otherwise,} \end{cases} \tag{2.22}$$

where d_M truncates the values according to the size of convolution filters. The configurations for learning a regressor are given by $\mathcal{D} = \{(f_i, z_i)\}_{i=1}^N$, where $f_i \in \mathbb{R}^n$ denotes feature vector of $\mathbf{x}_i$ on the image I and $z_i = d(\mathbf{x}_i)$. Specifically, the feature vector concatenates convolution filtering responses $f_i = [(\mathbf{F}_1 * I)(\mathbf{x}_i), \cdots, (\mathbf{F}_n * I)(\mathbf{x}_i)]^\intercal$, where the convolution filter banks are computed by [Rigamonti 12]. Gradient Boosting [Hastie 09] is used to learn a regression function. Finally, likewise Canny edge detector [Canny 86], Non Maximum Suppression (NMS) is performed along to the perpendicular direction of the structure for thinning the output function. Figure 2.8 provides the intermediate steps of SSDT algorithm proposed in [Sironi 14].



(a) I (b) C (c) $D_C(\mathbf{x})$ (d) $d(\mathbf{x})$ (e) $h(\mathbf{x})$ (f) NMS

Figure 2.8: Intermediate results of the learning based SSDT algorithm: (a) input image, (b) centerline, (c) distance transform, (d) a function to be used for learning, (e) output function of Gradient Boosting, and (f) applying NMS on the output function to localize centerlines. Images courtesy of [Sironi 14].

2.3 Conclusions

The previous works on curvilinear structure reconstruction modeling have been studied geometric features of the curvilinear structure on images. We review the curvilinear structure reconstruction algorithms to analyze blood vessel networks, road net-

works, biological images, facial wrinkles, and defects of materials. To summarize, the curvilinear structure locally exhibits thin, elongated, and tube-like structure on images. We summarize the implementation of curvilinear feature descriptors based on image gradient profiles and morphological profiles. Image gradient information is used to extract local image features which are analogous to elongated tubular structures and to estimate local orientation. A set of convolution filters is proposed to capture such geometric properties. Morphological profiles detect valid paths on a graph which reflects the topological feature of the curvilinear structure. We performs the morphological operation to find remote pixels on the same curvilinear structure within a pre-defined length.

Topology of the curvilinear structure varies depending on its application, so that different representation techniques are preferred to reconstruct the underlying curvilinear structures. Segmentation algorithms are apt to analyze irregular shaped curvilinear structure and provides a tubularity score for each pixel. Graph-based algorithms represent the latent curvilinear structure as a geodesic path on a graph. Stochastic image modeling reconstructs curvilinear structures as a set of disjoint line segments. Geometric prior defines the spatial interaction of the line segments. Machine learning based approaches is developed to analyze curvilinear structure not depending on a specific application.

In this dissertation, we aim to develop curvilinear structure reconstruction methods which are applicable to various types of datasets. In Chapter 3, we propose a stochastic model which employs image gradient profiles as the image based term and constrains spatial interaction of line segments. In Chapter 4, we develop a ranking learning system and graph based representation to infer curvilinear structures.

Chapter3

Chapter 3

Stochastic Modeling of Curvilinear Structure Reconstruction

Abstract

This chapter presents a novel stochastic model and the corresponding optimization technique to reconstruct curvilinear structures. We sample multiple line segments which are associated with the latent curvilinear structure. Therefore, we aim to maximize a posterior probability density with respect to a set of line segments for given image data. To fit the set of line segments into image data, we compute curvilinear feature descriptors based on oriented image gradient and morphological profiles. We employ support vector machine to estimate the distribution of the proposed curvilinear features and to regularize sampling space. We constrain local interaction of the line segments to obtain smoothly connected line configuration. The optimization technique consists of two steps to reduce the significance of the parameter selection in our stochastic model. We simulate several Markov chains with different parameter settings to collect line hypotheses on the same configuration space. Then, we maximize the consensus among line hypotheses to reduce the sampling space and to improve the reliability of the curvilinear structure reconstruction. Experimental results show that the algorithm effectively localizes curvilinear structures on a wide range of datasets.

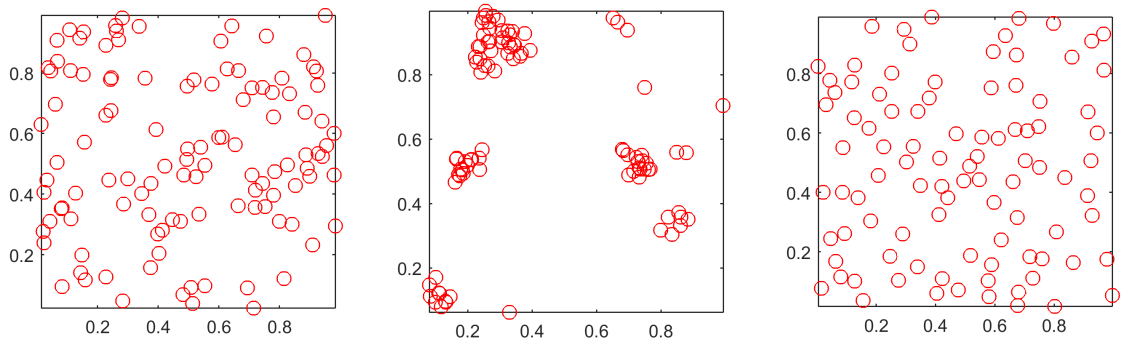
Contents

SAMPLING techniques with geometric priors have been exploited to detect multiple line segments in a scene [Lacoste 05, Tu 06, Schlecht 07, Batool 12, Predoehl 13, Jeong 14]. The *marked point process* (MPP) framework [Stoyan 87, Lieshout 00, Descombes 02, Chatelain 12] is a stochastic process which can enforce high level constraints on shape prior. However, the MPP model requires heavy formalization to interpret spatial distribution of the objects. Large number of parameters should be defined to describe the geometric shape of the objects (*modeling parameters*) and to control the relative importance of data and prior energy terms (*hyperparameters*). MPP modeling has been considered less practical to solve general problem because the performance is very sensitive for the selection of parameters. Although *stochastic expectation maximization* algorithm [Celeux 96, Møller 03] has been used to estimate modeling parameters, it exhibits both speed and scalability issues.

In this chapter, we propose a novel stochastic model based on MPP framework for curvilinear structure extraction in a fully automatic way, where the performance is not biased by the hyperparameter selection. Indeed, our stochastic model can detect wide types of input data without a sophisticated parameter tuning process. In Section 3.1, we review MPP framework [Stoyan 87, Lieshout 00] to provide a mathematical description of the proposed model. To fit into image data, we proposed curvilinear features with rotated image gradient and morphological profiles. Support vector machine (SVM) [Vapnik 95] is employed to approximate the data likelihood distribution of the given proposed curvilinear features. The prior distribution is also defined on local configuration to implement smooth connection among line segments (Section 3.2). Markov chain Monte Carlo (MCMC) sampler [Gilks 95, Green 95, Robert 04] with delayed rejection scheme [Green 01] is employed to optimize the proposed probability density function. We explain the implementation details of transition kernels for MCMC sampler in Section 3.3. To avoid the burden of hyperparameter selection, we generate multiple candidates of the line configuration with different hyperparameter settings. Next, we combine the whole set of line candidates in a way that maximizes the consensus among line detection results (Section 3.4). Extensive experiments on various datasets including facial wrinkles, road cracks, DNA filaments, and blood vessels demonstrate the effectiveness of the proposed model for extracting thin curvilinear structures (Section 3.5).

3.1 Revisited Marked Point Process Framework

Modeling with spatial point process aims to describe the spatial pattern of data using a finite number of parameters. We evaluate the spatial point patterns with respect to *randomness* in a finite space. For example, we employ a statistical model to



(a) Completely random (b) Attraction (c) Repulsion

Figure 3.1: A spatial point process is a mathematical model to analyze spatial point patterns using a finite number of parameters. More precisely, the spatial point process is a random variable which maps an unordered set of points in a finite space. We can compare different point patterns according to the randomness in the space.

realize a completely independent point pattern. Then, we observe how the given points are differently (or similarly) distributed in space comparing to the randomly generated point patterns (see Figure 3.1).

Formally speaking, a spatial point process X is a random variable and its realization is associated with an unordered set of points $\Omega_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in a compact set $\mathcal{F} \subset \mathbb{R}^d$ in d-dimensional space. A spatial point process on $\mathcal{F}$ maps from a measurable space $(\mathcal{F}, \mathcal{B}, \nu)$ onto the set of configurations $\Omega = \cup_{n=0}^{\infty} \Omega_n$, where $\mathcal{B}$ denotes σ-algebra of B, and ν is a positive counting measure on $\mathcal{F}$. In other words, for all bounded Borel set $B \subseteq \mathcal{F}$, the number of points $N_X(B)$ falling in B is a finite random variable.

Furthermore, for each point we add an additional information to describe the shape of an object. Let $s_i = (\mathbf{x}_i, \mathbf{m}_i)$ be an augmented point (object) which encodes its center location $\mathbf{x}_i \in \mathcal{F}$ and a shape information (mark) $\mathbf{m}_i \in \mathcal{M}$. Similarly to sample a point with probability measure μ, we obtain a parameter to describe its shape with a probability measure μ_M. Thus, a realization of the *marked point process* on $\Psi = \mathcal{F} \times \mathcal{M}$ is a finite random configuration set $\mathbf{s} = \{s_1, \ldots, s_n\} \in \Psi$.

In practice, the *homogeneous Poisson process* is employed to develop a stochastic model to quantify randomly scattered point patterns. The homogeneous Poisson process X with a positive intensity $\nu > 0$ satisfies the following properties:

Property 1. For all bounded Borel sets B, the number of points $N_X(B)$ falling in B

follows a discrete Poisson distribution with mean $\nu|B|$:

$$\Pr(N_X(B) = k) = \frac{\nu|B|^k}{k!} e^{-\nu|B|}.$$

Property 2. For disjoint bounded sets, $B_1, \ldots, B_k$, the corresponding random variables $N_X(B_1), \ldots, N_X(B_k)$ are independent.

Straightforwardly, the Poisson process can be generalized by a non-constant intensity measure $\nu(\cdot)$ on $\mathcal{F}$. This process is called an *inhomogeneous Poisson process*. The Poisson processes realize independently scattered points on the space, while real objects can be interactive on a local configuration. For instance, we want to avoid overlapping between objects and to regularize the orientations. To take into account interactions of the scattered points, we specify a density $f(\cdot)$ with respect to the homogeneous Poisson process with intensity measure $\nu(\cdot)$. Let $f(\cdot)$ be a non-negative function on sampling space. Then, the measure $\boldsymbol{\nu}(\cdot)$ having density $f(\cdot)$ with respect to $\nu(\cdot)$ is defined by:

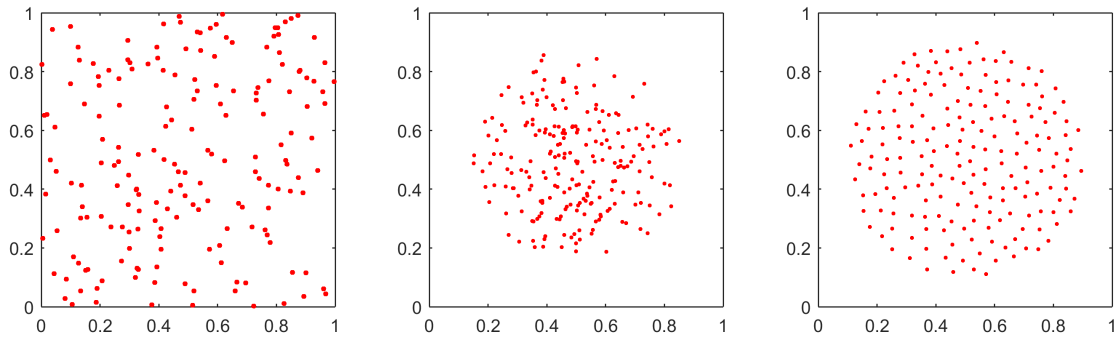
$$\boldsymbol{\nu}(B) = \int_B f(\mathbf{s})\nu(d\mathbf{s}). \tag{3.1}$$

For example, Markov point process defines $f(\cdot)$ with the pairwise interaction (repulsion or attraction) of points according to their distance. Figure 3.2 compares the realization of homogeneous Poisson process, inhomogeneous Point process, and repulsive Markov point process, respectively.

Now we design the density function $f(\cdot)$ which can control point pattern according to the given data and prior knowledge of their physical properties. The density function also refers to a probability distribution of the points. In this chapter, we represent a line segment $s_i = (\mathbf{x}_i, \ell_i, \theta_i)$ as a tuple of its center location, length, and orientation, respectively. To reconstruct the curvilinear structure, we collect smoothly connected line segments on the configuration space Ψ. In other words, our goal is to find an optimal set of line segments $\hat{\mathbf{s}}$ which maximizes posterior probability for given image $f(\mathbf{s}) \simeq p(\mathbf{s}|I)$. The posterior probability can be separated into data likelihood $p(I|\mathbf{s})$ and prior probability $p(\mathbf{s})$ such as:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\Psi}{\operatorname{argmax}}\, p(\mathbf{s}|I) = \underset{\mathbf{s}\in\Psi}{\operatorname{argmax}}\, p(I|\mathbf{s})p(\mathbf{s}). \tag{3.2}$$

In general, Markov chain Monte Carlo (MCMC) samplers [Green 95, Green 01, Gamal-Eldin 10, Verdié 14] are employed to maximize the proposed density function $p(\mathbf{s}|I)$. Each state of a discrete Markov chain $(X_t)_{t\in\mathbb{N}}$ corresponds to a random configuration on the Ψ. The chain is locally perturbed by transition kernels, and is evolved to converge to the stationary distribution which is identical to the maximum of the



(a) Homogeneous Poisson process (b) Inhomogeneous Poisson process (c) Repulsive Markov point process

Figure 3.2: Comparison with the realization of point patterns obtain by (a) homogeneous Poisson process, (b) inhomogeneous Poisson process, and (c) repulsive Markov point process. For a finite space, homogeneous Poisson process employs a constant intensity measure, whereas inhomogeneous Poisson process employs intensity measure as a function. Both Poisson processes ignore correlation of points in the space. On the other hand, Markov point process constrains the local interaction of neighboring points using a non-negative density function. Here, repulsive constraint prevents overlapping between points.

proposed probability density. In Section 3.3, we further explain MCMC dynamics and implementation of the transition kernels.

3.2 Stochastic modeling of curvilinear structure

In this section, we propose a stochastic model for curvilinear structure reconstruction. The stochastic model consists of data likelihood $p(I|\mathbf{s})$ to localize a set of the line segments $\mathbf{s}$ for given image data I and prior distribution $p(\mathbf{s})$ to define local geometry of line segments. Let $z_i \in \{-1, 1\}$ denote the latent variable for pixel $\mathbf{x}_i$; $z_i = 1$ if $\mathbf{x}_i$ is on the curvilinear structure, otherwise $z_i = -1$. We predict a pixel $\mathbf{x}_i$ whether it belongs to the curvilinear structure using the corresponding curvilinear feature vector $\phi(\mathbf{x}_i)$. Specifically, we propose the curvilinear feature descriptors $\phi : I \mapsto \mathbb{R}^N$ based on image gradient or morphological profiles, where N represents the dimension of the feature space. The data likelihood combines information of pixels on the given line segment:

$$p(I|s_i) = \frac{1}{Z} \exp\Big(- E_{\text{data}}(s_i)\Big) = \frac{1}{Z} \exp\left(- \sum_{\mathbf{x}_j \in s_i} E_{\text{data}}(\phi(\mathbf{x}_j))\right), \tag{3.3}$$

where E_{data} denote Gibbs energy form and $Z = \int_{s \in \mathbf{s}} \exp\Big(-E_{\text{data}}(s)\Big)$ is a normalizing constant.

(a) $s_m \sim s_n$ (b) $s_m \not\sim s_n$

Figure 3.3: Line segments (s_m, s_n) interact if they are close enough.

To compute prior probability, we assume that an interaction of the line segments is valid if and only if they are close enough in the image coordinate. Thus, we consider pairs of line segments such that their center distance is smaller than half the sum of their lengths:

$$s_m \sim s_n = \left\{ (s_m, s_n) \in \Psi^2 : 0 < \|\mathbf{x}_m - \mathbf{x}_n\|_2 \leq \frac{\ell_m + \ell_n}{2} + \epsilon \right\}, \tag{3.4}$$

where ϵ denotes the marginal distance to be connected with each other. With the consideration of neighboring condition (3.4), we can compute the prior probability as follows:

$$p(\mathbf{s}) = p(s_1, \ldots, s_{\#(\mathbf{s})}) = \prod_{s_m \sim s_n} p(s_m | s_n), \tag{3.5}$$

where $\#(\mathbf{s})$ denotes the number of line segments in the configuration $\mathbf{s}$.

The probability density is interchangeable with Gibbs energy form as a negative log probability [Gibbs 02]: $E(x) \propto -\log p(x)$. We denote energy form of data likelihood probability as E_{data} and prior probability as E_{prior}. Therefore, maximizing the proposed posterior probability is equivalent to minimizing the total energy of line configuration:

$$\hat{\mathbf{s}} = \underset{\mathbf{s} \in \Psi}{\operatorname{argmax}}\, p(\mathbf{s}|I) = \underset{\mathbf{s} \in \Psi}{\operatorname{argmax}} \prod_{i=1}^{\#(\mathbf{s})} p(s_i|I) \tag{3.6}$$

$$= \underset{\mathbf{s} \in \Psi}{\operatorname{argmax}} \prod_{i=1}^{\#(\mathbf{s})} p(I|s_i) \prod_{s_m \sim s_n} p(s_m|s_n) \tag{3.7}$$

$$= \underset{\mathbf{s} \in \Psi}{\operatorname{argmin}} \sum_{i=1}^{\#(\mathbf{s})} E_{\text{data}}(s_i) + \sum_{s_m \sim s_n} E_{\text{prior}}(s_m, s_n), \tag{3.8}$$

3.2.1 Data likelihood

Data likelihood employs a curvilinear feature descriptor to measure the fidelity of the pixel with respect to the given image data. First, we exploit rotated image gradient magnitudes. A convolution filter $\mathbf{F}_\theta$ is applied to extract tubular structures oriented to the corresponding direction θ. The rotated image gradient information is suitable to detect local tubular structure of the image; however, image gradient often fails to classify pixels even though they are remotely connected in the same curvilinear structure. To make up for such failure case, we propose a curvilinear feature descriptor using morphological profiles. We then combine several feature maps which highlights the latent curvilinear structure with different statistics. Hence, Support vector machine (SVM) [Vapnik 95] is employed to obtain relative importance of each feature map and to approximate the data likelihood distribution of the given curvilinear feature vector.

Curvilinear feature based on image gradient statistics

We perceive the thin linear shape structure due to a contrast dissimilarity with the background textures. Specifically, pixels on the curvilinear structure are slightly darker or brighter than its surroundings, and spread on a similar direction. Thus, we analyze image gradient since it can measure local intensity variation. We compute the rotated gradient information by convolving the given input image I with rotatable filters $\{\mathbf{F}_\theta\}$, which are defined in the previous chapter (Section 2.1). Let $\nabla_\theta I = \mathbf{F}_\theta * I$ denote the image gradient map which accentuates image gradient magnitude corresponding to the orientation θ. When the input image is heavily corrupted by noise or composed of uneven textures, observing gradient distribution often fails to detect linear structures. To ease this problem, we also assume that the intensity on the line segments is homogeneous. We then compute a curvilinear feature vector $\phi_1(\mathbf{x})$ for pixel $\mathbf{x}$ by combining these information :

$$\phi_1(\mathbf{x}) = [I(\mathbf{x}),\ \nabla_\theta I(\mathbf{x})]^\intercal, \tag{3.9}$$

Curvilinear feature based on morphological profiles

Inner similarity of pixels on the curvilinear structure is hard to get using local gradient features due to irregular illuminate condition, rough background texture, and noise. Morphological operator detects linear shape features using recursive structural patterns on the image. Hence, we employ path opening operations [Talbot 07] by controlling the length parameter to obtain different feature maps which can compensate the drawbacks of image gradient features. We assume that pixels belonging to the curvilinear structure are brighter than their neighborhoods. Otherwise, we invert the intensity level of the input image before applying the morphological operator. We then

denote the curvilinear feature vector $\phi_2(\mathbf{x})$ which combines the morphological filtering responses I_L with increasing order of length parameter L. That is

$$\phi_2(\mathbf{x}) = \left[I_{L_1}(\mathbf{x}), \ldots, I_{L_4}(\mathbf{x})\right]^\intercal, \quad \text{where } L_1 < \cdots < L_4.$$

Initial classification of the latent curvilinear structure

We concatenate two curvilinear feature vectors $\phi_1(\mathbf{x})$ and $\phi_2(\mathbf{x})$ to compute data likelihood of being curvilinear structure: $\phi(\mathbf{x}) = \begin{bmatrix}\phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x})\end{bmatrix}$. In this chapter we employ the SVM for the initial classification of the latent curvilinear structure. Let $\mathcal{D} = \{(\phi(\mathbf{x}_i), z_i)\}_{i=1}^K$ be a training set. In practice, we randomly select 50 training data points for positive and negative samples, respectively. To obtain a linear hyperplane, we solve following quadratic programming problem:

$$\begin{array}{ll} \min\limits_{b,\mathbf{w},\xi} & \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{K}\sum_{i=1}^K \xi_i \\ \text{subject to} & z_i(\mathbf{w}^\intercal\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i; \quad \text{for } i = 1, \ldots, K \end{array} \tag{3.10}$$

where K is the total number of training data, b represents the bias of the hyperplane, and $\xi \geq 0$ denotes slack variables for regression. The constant value C controls the relative importance of regularization ($\frac{1}{2}\mathbf{w}^\intercal\mathbf{w}$) and regression ($\frac{C}{K}\sum_{i=1}^K \xi_i$) terms (please refer to Chapter 4 for more detailed description of SVM).

The solution of (3.10) $\mathbf{w}$ defines the decision function $h_i = h(\phi(\mathbf{x}_i)) = \mathbf{w}^\intercal\phi(\mathbf{x}_i) + b$ of the SVM classifier, and estimate the latent label $\hat{z}_i = \texttt{sign}(h_i)$. The distance from the decision boundary (hyperplane) to a feature vector can be considered as a psuedo-probability belonging to curvilinear structure, which can be approximated by a sigmoid function [Platt 00]:

$$p_{\mathbf{w}}(z_i = 1|\phi(\mathbf{x}_i)) = \frac{1}{1 + \exp(\alpha h_i + \beta)}, \tag{3.11}$$

where regularization parameters (α and β) are estimated by maximum likelihood. We want to regulate that data likelihood probability is likely to the initial estimation obtained via SVM classifier. Moreover, we assume that the data likelihood of the given curvilinear feature vectors follows Gaussian distribution with mean $\boldsymbol{\mu}^{\mathrm{d}}$ and covariance $\boldsymbol{\Sigma}^{\mathrm{d}}$:

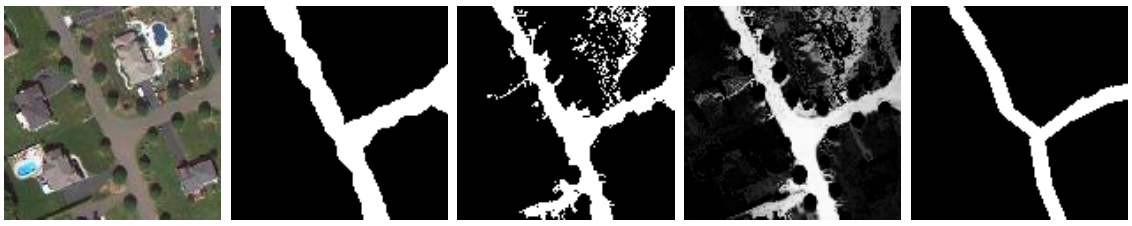
$$\boldsymbol{\mu}^{\mathrm{d}} = \frac{\sum_i p(z_i = 1|\phi(\mathbf{x}_i))\phi(\mathbf{x}_i)}{\sum_i p(z_i = 1|\phi(\mathbf{x}_i))}, \qquad \boldsymbol{\Sigma}^{\mathrm{d}} = \left(\mathbf{w}\mathbf{w}^\intercal\right)^{\frac{1}{2}}. \tag{3.12}$$

We then define the data likelihood for given curvilinear feature vector $\phi(\mathbf{x})$:

$$E_{\text{data}}(\phi(\mathbf{x}_i)) = -\omega_d \log p_{\mathbf{w}}(z_i = 1|\phi(\mathbf{x}_i)) - \log \mathcal{N}(\phi(\mathbf{x}_i); \boldsymbol{\mu}^{\text{d}}, \boldsymbol{\Sigma}^{\text{d}}), \tag{3.13}$$

$$\simeq \omega_d \exp(\alpha h_i + \beta) + \left(\phi(\mathbf{x}_i) - \boldsymbol{\mu}^{\text{d}}\right)^{\intercal} \boldsymbol{\Sigma}^{\text{d}^{-1}} \left(\phi(\mathbf{x}_i) - \boldsymbol{\mu}^{\text{d}}\right) \tag{3.14}$$

where ω_d denotes a hyperparameter to support the initial classification results of SVM.



(a) Input (b) Ground truth (c) Binary classification ($\hat{z}_i$) (d) Relaxed probability ($p_{\mathbf{w}}$) (e) Proposed

Figure 3.4: For the input image (a), we compare the manually labeled result (b), outcome of the binary SVM classifier (c) and the posterior probabilities (d). The proposed stochastic model (e) refines the weak points of the SVM classifier by sampling line segments with geometry prior constraints.

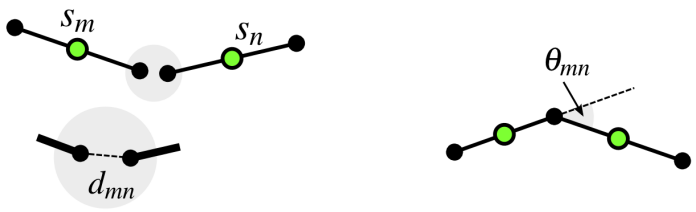
The binary SVM classifier fails to classify the curvilinear structure so that it yields jittering artifacts around boundaries. This is because the SVM classifier assumes identical-independent-distributed (i.i.d.) condition for the sample points when it learns hypothesis. To improve the classification performances, we employ prior constraints in a local line configuration (see Figure 3.4).

3.2.2 Prior Energy

In this section, we propose the prior energy to define spatial interactions of a local line configuration. We define the coupling feature vector $\mathbf{c}_{mn}$ for adjacent line segments $s_m \sim s_n$ by neighboring condition (3.4). The coupling feature vector takes into account the connectivity and curvature of pairwise line segments, such as:

$$\mathbf{c}_{mn} = [d_{mn},\ \theta_{mn}]^{\intercal}, \tag{3.15}$$

where d_{mn} denotes the minimum distance between endpoints of s_m and s_n, which is related to the connectivity. θ_{mn} is the angle difference to measure the curvature of two line segments (see Figure 3.5). Intuitively, small values of d_{mn} and θ_{mn} are desired to obtain connected and aligned line configuration: $\mathbf{c}_{mn}^{\intercal}\mathbf{c}_{mn} \simeq 0$.

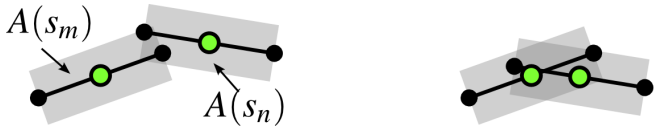


(a) (b)

Figure 3.5: For coupling feature $\mathbf{c}_{mn}$ We measure (a) the distance of adjacent line segments d_{mn} and (b) the angle differnece between line segments θ_{mn}.

We also compute the overlapping area $\Upsilon(s_m, s_n)$ to reject congestion of lines within a local area. In order to evaluate an overlapping area between line segments, we dilate the line segments with a three pixel-radius disk, and then count the number of pixels falling in the same area. Suppose that we have a set of points $A(s_m)$ which is a dilated version of the line segment s_m, and $|A(s_m)|$ denotes the number of pixels in $A(s_m)$. As shown in Figure 3.6, we penalize a configuration $\{s_m, s_n\}$, when a portion of the overlapping area is greater than 10% of $\min\{|A(s_i)|, |A(s_j)|\}$. The criteria for rejection are then given as

$$\Upsilon(s_m, s_n) = \begin{cases} 0 & \text{if } \frac{|A(s_i) \cap A(s_j)|}{\min\{|A(s_i)|, |(A(s_j)|\}} < 0.1, \\ 1 & \text{otherwise.} \end{cases} \tag{3.16}$$



(a) $\Upsilon(s_m, s_n) = 0$ (b) $\Upsilon(s_m, s_n) = 1$

Figure 3.6: To prevent congestion of line segments, we compute the overlapping area of pairwise line segments. We take into account either pixels on the line segment s_m and its surroundings $A(s_m)$, which are denoted by gray color in this figure. The criteria function $\Upsilon(s_m, s_m)$ computes the number of pixels falling into the overlapping area of two line segments. We allow small overlapping between two line segments (a), whereas we reject line configuration (b).

Assuming that $\mathbf{c}_{mn}$ follows a mixture of Gaussian distributions with zero mean $\mathbf{0}$

and covariance $\mathbf{\Sigma}^{\mathrm{p}}$, we can write the prior probability:

$$E_{\text{prior}}(s_m, s_n) = \omega_p \Upsilon(s_m, s_n) - \sum_{d=1}^{D} \boldsymbol{\pi}_d^{\mathrm{p}} \log \mathcal{N}(\mathbf{c}_{mn}; \mathbf{0}, \mathbf{\Sigma}_d^{\mathrm{p}}), \tag{3.17}$$

$$\simeq \omega_p \Upsilon(s_m, s_n) + \sum_{d=1}^{D} \boldsymbol{\pi}_d^{\mathrm{p}} \mathbf{c}_{mn}^{\intercal} \mathbf{\Sigma}_d^{\mathrm{p}-1} \mathbf{c}_{mn}, \tag{3.18}$$

$$\simeq \omega_p \Upsilon(s_m, s_n) + \pi_1 d_{mn}^2 + \pi_2 \theta_{mn}^2, \tag{3.19}$$

where $D = 2$ denotes the number of Gaussian mixtures, and $\boldsymbol{\pi}_d^{\mathrm{p}}$ is the corresponding weighting coefficient. ω_p is a hyperparameter which controls the rejection rate.

The hyperparameters of data likelihood ω_d and prior probability $\{\boldsymbol{\pi}_d^{\mathrm{p}}\}_{d=1}^{D}$ and ω_p play important roles to sample reliable line segments. For example, ω_d penalizes the birth of a line segment in the area which is not associated with the initial estimation. $\{\boldsymbol{\pi}_d^{\mathrm{p}}\}_{d=1}^{D}$ promotes adjacent line segments to be connected and aligned in the final configuration. ω_p supports the sensitivity of overlapping criteria. However, it is hard to estimate such hyperparameters because of implicit dependencies among them. Specifically, for the prior distribution, we cannot exploit true statistics of related parameters for learning phase. We will tackle this issues in section 3.4.

3.3 Optimization

To maximize the proposed probability density of line segments, we employ a Markov chain Monte Carlo (MCMC) sampler. In this section we discuss how the MCMC sampling devotes to maximize the probability density function in a complex sample space and its implementation details.

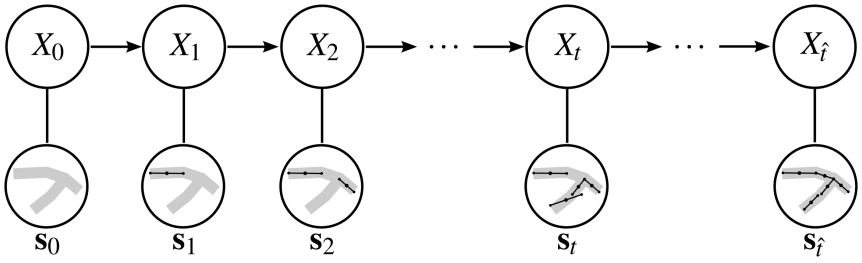


Figure 3.7: The latent curvilinear structure, denoted by gray color, is reconstructed by a set of line segments. A state of the Markov chain is related to a line configuration, and updates its state according to the transition kernels. We finally obtain an equilibrium state of the Markov chain which maximizes the corresponding probability density in the sampling space.

3.3.1 Markov chain Monte Carlo sampler

We employ a discrete Markov chain $\{X_t\}_{t\in\mathbb{N}}$ to support the proposed probability density of line segments in the configuration space Ψ. A state of the Markov chain X_t at time t is a random variable which is associate with a set of line segments $\mathbf{s}_t$ (see Figure 3.7). We simulate a discrete Markov chain using a transition kernel $\mathbf{K}(\mathbf{s}_a, \mathbf{s}_b) = \Pr(X_b = \mathbf{s}_b | X_a = \mathbf{s}_a)$ which updates state from X_a to X_b. To obtain an unique stationary probability $p(\mathbf{s}_b|I)$, the Markov chain shall satisfy *ergodicity* [Chatelain 12]:

$$p(\mathbf{s}_b|I) = \sum_{\mathbf{s}_a\in\Psi} p(\mathbf{s}_a|I)\mathbf{K}(\mathbf{s}_a, \mathbf{s}_b), \quad \forall \mathbf{s}_b \in \Psi. \tag{3.20}$$

Thus, the Markov chain eventually reaches an equilibrium state which maximizes the proposed density in the configuration space. In practice, this property (3.20) can be implied by the *detailed balance condition* [Green 95, Chatelain 12] for any $\mathbf{s}_a \neq \mathbf{s}_b \in \Psi$:

$$p(\mathbf{s}_a|I)\mathbf{K}(\mathbf{s}_a, \mathbf{s}_b) = p(\mathbf{s}_b|I)\mathbf{K}(\mathbf{s}_b, \mathbf{s}_a) \tag{3.21}$$

The transition kernel consists of several sub-transition kernels which defines different types of jumps to perturb the state:

$$\mathbf{K}(\mathbf{s}_a, \mathbf{s}_b) = \sum_{\mathrm{m}} q_{\mathrm{m}}\mathbf{K}_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b), \quad \sum_{\mathrm{m}} q_{\mathrm{m}} = 1, \tag{3.22}$$

where q_{m} denotes a probability to choose m-th type of sub-transition kernel $\mathbf{K}_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b)$. The sub-transition kernel is separated into a proposal probability $\mathbf{Q}_{\mathrm{m}}(\mathbf{s}_b|\mathbf{s}_a)$ for moving from $\mathbf{s}_a$ to $\mathbf{s}_b$ and the corresponding acceptance ratio $\alpha_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b)$. That is $\mathbf{K}_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b) = \alpha_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b)\mathbf{Q}_{\mathrm{m}}(\mathbf{s}_b|\mathbf{s}_a)$. With holding the detailed balance condition (3.21), we compute the acceptance ratio $\alpha_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b)$ as follows [Hastings 70]:

$$\alpha_{\mathrm{m}}(\mathbf{s}_a, \mathbf{s}_b) = \min\left(1, \frac{\mathbf{Q}_{\mathrm{m}}(\mathbf{s}_b|\mathbf{s}_a)p(\mathbf{s}_b|I)}{\mathbf{Q}_{\mathrm{m}}(\mathbf{s}_a|\mathbf{s}_b)p(\mathbf{s}_a|I)}\right). \tag{3.23}$$

In the following section, we design the proposal kernels which efficiently perturb the line configuration in order to maximize the proposed probability density.

3.3.2 Reversible Jump Markov chain Monte Carlo sampler

Since the dimensionality of the line configuration can be changed during simulation process, we therefore employ *Reversible jump Markov chain Monte Carlo* (RJMCMC) sampler [Green 95] to maximize the probability density function. The RJMCMC sampler is an iterative method that locally perturbs a current configuration $\mathbf{s}$ with a transi-

tion kernel, *i.e., birth-and-death* (BD) and *linear transform* (LT). Each sub-transition kernel consists of pair of reversible jumps and holds the detailed balance condition (3.21). Acceptance ratio $\alpha_{\mathrm{m}}(\mathbf{s}, \mathbf{s}')$ is compared with a stochastic value $\texttt{rand}[0, 1]$ to update a new configuration $\mathbf{s}'$. The RJMCMC sampler is coupled with the *simulated annealing* (SA) algorithm [Kirkpatrick 83] to secure the convergence of the Markov chain via relaxation parameter T (temperature); the temperature gradually decreases as the iteration goes on. To compute an acceptance ratio of the transition kernel, we use a density $p(\mathbf{s}|I)^{1/T}$ instead of $p(\mathbf{s}|I)$. The acceptance ratio is

$$\alpha_{\mathrm{m}}(\mathbf{s}, \mathbf{s}') = \min\left(1, \frac{\mathbf{Q}_{\mathrm{m}}(\mathbf{s}|\mathbf{s}')}{\mathbf{Q}_{\mathrm{m}}(\mathbf{s}'|\mathbf{s})} \frac{p(\mathbf{s}'|I)^{1/T}}{p(\mathbf{s}|I)^{1/T}}\right). \tag{3.24}$$

Algorithm 1 summarizes the optimization process using RJMCMC.

Algorithm 1 RJMCMC sampler [Green 95]

Initialize: $X_0 \leftarrow \mathbf{s}_0$ (or $X_0 \leftarrow \emptyset$), $t \leftarrow 0$, $T \leftarrow T_0$
while $T > T_{\min}$ **do**
  $\mathbf{s} \leftarrow X_t$
  Choose a transition kernel $\mathbf{K}_{\mathrm{m}}$ according to probability q_m
  Propose a new configuration $\mathbf{s}'$ with $\mathbf{Q}_{\mathrm{m}}(\mathbf{s}'|\mathbf{s})$
  if $\alpha_{\mathrm{m}}(\mathbf{s}, \mathbf{s}') > \texttt{rand}[0, 1]$ **then**
    $X_{t+1} \leftarrow \mathbf{s}'$
  end if
  $t \leftarrow t + 1$
  Decrease the temperature: $T \leftarrow T_t$
end while

More precisely, the **BD kernel** changes the dimensionality of the current configuration $\mathbf{s}$ by adding a new line segment or removing an existing line segment. When the birth kernel proposes a new configuration $\mathbf{s}' = \mathbf{s} \cup \{s\}$, the length and the orientation of the new line segment are uniformly sampled from the mark space $M = [\ell_{\min}, \ell_{\max}] \times [\theta_{\min}, \theta_{\max}]$, where $\ell_{\min}$ and $\ell_{\max}$ are the minimum and maximum length of the line segment, respectively. Similarly, $\theta_{\min}$ and $\theta_{\max}$ denote the minimum and maximum orientation of the line segment, respectively. Note that we refuse a birth of the line segment lying on *singular points*, which have zero gradient magnitudes. On the other hand, the death kernel removes a line segment which is randomly picked from the current configuration. Thus, a new configuration $\mathbf{s}' = \mathbf{s} \backslash \{s\}$ is proposed by the death kernel. We compute the acceptance ratio of the birth kernel α_{B} and the death

kernel α_{D}, given by

$$\alpha_{\mathrm{B}}(\mathbf{s}, \mathbf{s}') = \min\left(1, \frac{p_{\mathrm{D}}}{p_{\mathrm{B}}} \frac{\mu_{\mathrm{M}}(M)}{\#(\mathbf{s})+1} \frac{p\left(\mathbf{s}'|I\right)^{1/T}}{p(\mathbf{s}|I)^{1/T}}\right), \tag{3.25}$$

$$\alpha_{\mathrm{D}}(\mathbf{s}, \mathbf{s}') = \min\left(1, \frac{p_{\mathrm{B}}}{p_{\mathrm{D}}} \frac{\#(\mathbf{s})}{\mu_{\mathrm{M}}(\mathcal{M})} \frac{p(\mathbf{s}'|I)^{1/T}}{p\left(\mathbf{s}|I\right)^{1/T}}\right), \tag{3.26}$$

where $\mu_{\mathrm{M}}(\cdot)$ is Lebesgue measure for marked space $\mathcal{M}$. That is $\mu_{\mathrm{M}}(\mathcal{M}) = (\ell_{\max} - \ell_{\min})(\theta_{\max} - \theta_{\min})$.

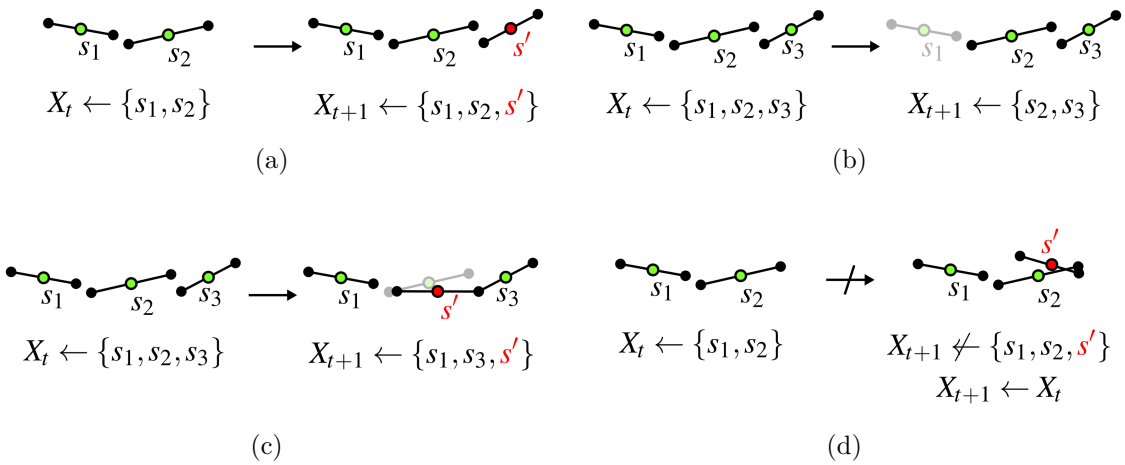
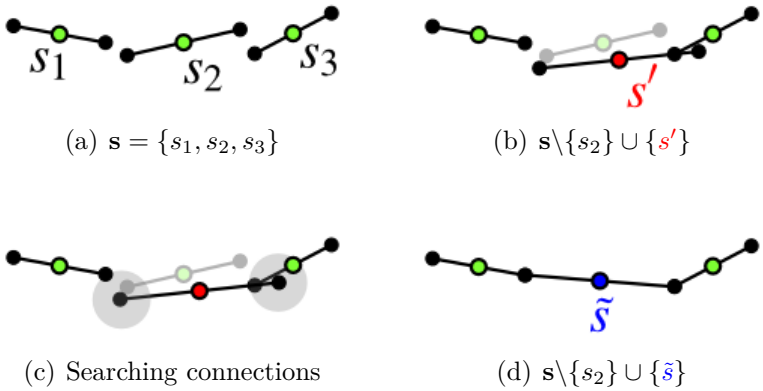


Figure 3.8: The MCMC sampler perturbs the current configuration $\mathbf{s}$ via sub-transition kernels: (a) birth kernel, (b) death kernel, and (c) linear-transform kernel. Schematic diagrams illustrate how the sub-transition kernels change topology of the corresponding configuration, and a state of the Markov chain is gradually updated in order to maximize the proposed probability density function. If a new configuration disagrees with the probability density function, the chain will remain at the previous state as shown in (d).

The **LT kernel** chooses a line segment s randomly, and then modifies its model parameters: $s = (\mathbf{x}, (\ell, \theta)) \rightarrow s' = (\mathbf{x} \pm \Delta\mathbf{x}, (\ell \pm \Delta\ell, \theta \pm \Delta\theta))$, where $\Delta\mathbf{x}$, $\Delta\ell$, and $\Delta\theta$ denote changes of center position, length, and orientation, respectively. The LT kernel draws a new configuration $\mathbf{s}' = \mathbf{s} \backslash \{s\} \cup \{s'\}$. The acceptance ratio of the LT kernel is defined by

$$\alpha_{\mathrm{LT}}(\mathbf{s}, \mathbf{s}') = \min\left(1, \frac{p(\mathbf{s}'|I)^{1/T}}{p(\mathbf{s}|I)^{1/T}}\right). \tag{3.27}$$



(a) $\mathbf{s} = \{s_1, s_2, s_3\}$ (b) $\mathbf{s}\backslash\{s_2\} \cup \{s'\}$

(c) Searching connections (d) $\mathbf{s}\backslash\{s_2\} \cup \{\tilde{s}\}$

Figure 3.9: Given configuration (a), if a line segment s' proposed by LT kernel is rejected (b), the delayed rejection kernel searches for the nearest extremes in the rest of line segments (c). An alternative line segment $\tilde{s}$, which enforces connectivity, will be proposed by interpolation of the retrieved points (d).

Figure 3.8 shows how the sub-transition kernels update the corresponding configurations.

3.3.3 Delayed Rejection

The LT kernel can be extended by the delayed rejection scheme [Green 01]. In order to reduce the burn-in time, we add heuristics to design the delayed rejection kernel. The main idea of the delayed rejection scheme is to give a second chance to a rejected configuration by enforcing connectivity of the line segments. The acceptance ratio of delayed rejection is defined by

$$\begin{aligned} \alpha^2_{\mathrm{LT}}(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) &= \min\left(1, \frac{\mathbf{Q}_{\mathrm{LT}}\left(\mathbf{s}'|\tilde{\mathbf{s}}\right)}{\mathbf{Q}_{\mathrm{LT}}\left(\mathbf{s}'|\mathbf{s}\right)} \frac{\mathbf{Q}^2_{\mathrm{LT}}\left(\mathbf{s}|\mathbf{s}', \tilde{\mathbf{s}}\right)}{\mathbf{Q}^2_{\mathrm{LT}}\left(\tilde{\mathbf{s}}|\mathbf{s}, \mathbf{s}'\right)} \frac{\left[1 - \alpha_{\mathrm{LT}}(\tilde{\mathbf{s}}, \mathbf{s}')\right]}{\left[1 - \alpha_{\mathrm{LT}}(\mathbf{s}, \mathbf{s}')\right]} \frac{p(\tilde{\mathbf{s}}|I)^{1/T}}{p(\mathbf{s}|I)^{1/T}}\right), \\ &\simeq \min\left(1, \frac{p(\tilde{\mathbf{s}}|I)^{1/T} - p(\mathbf{s}'|I)^{1/T}}{p(\mathbf{s}|I)^{1/T} - p(\mathbf{s}'|I)^{1/T}}\right). \end{aligned} \tag{3.28}$$

where $\mathbf{s}' = \mathbf{s}\backslash\{s\} \cup \{s'\}$, $\tilde{\mathbf{s}} = \mathbf{s}\backslash\{s\} \cup \{\tilde{s}\}$, and $\mathbf{Q}^2_{\mathrm{LT}}\left(\tilde{\mathbf{s}}|\mathbf{s}, \mathbf{s}'\right)$ is the transition kernel for the delayed rejection. Intuitively, an alternative configuration $\tilde{\mathbf{s}}$ is accepted if the corresponding probability $p(\tilde{\mathbf{s}}|I)$ is higher than $p(\mathbf{s}'|I)$. At the same time, it satisfies $p(\mathbf{s}'|I) < p(\mathbf{s}|I)$, which is a good reason to reject the intermediate configuration $\mathbf{s}'$. To propose an alternative line segment $\tilde{s}$, we look for the closest endpoints from both ends of s', which is rejected from the first trial. The line segment $\tilde{s}$ is generated

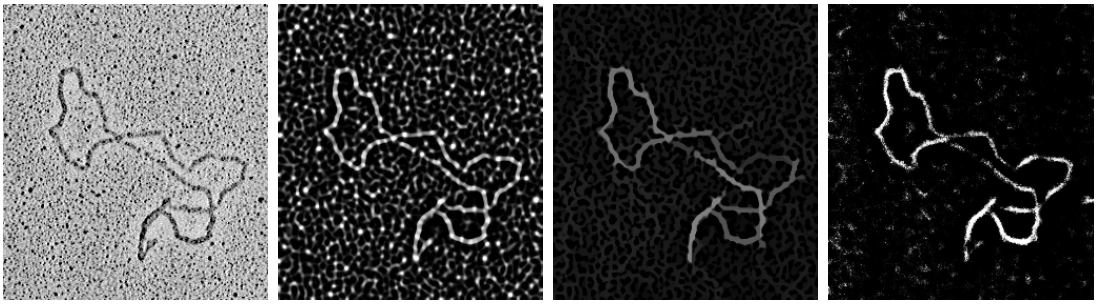
by interpolation of the retrieved points; we force the connectivity of the neighboring segments, so that a probability of being accepted increases in terms of prior distribution. Figure 3.9 summarizes the process of the delayed rejection kernel, and Algorithm 2 provides the pseudo-code of the RJMCMC sampler with delayed rejection.

Algorithm 2 RJMCMC sampler with delayed rejection

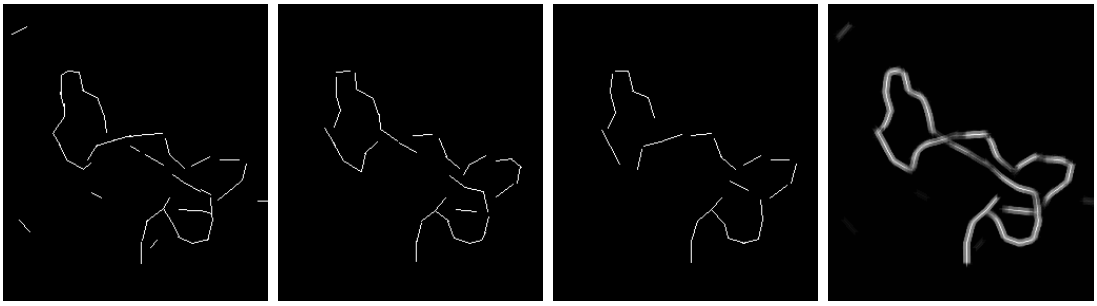
Initialize: $X_0 \leftarrow \mathbf{s}_0$ (or $X_0 \leftarrow \emptyset$), $t \leftarrow 0$, $T \leftarrow T_0$
while $T > T_{\min}$ **do**
  $\mathbf{s} \leftarrow X_t$
  Choose a transition kernel $\mathbf{K}_{\mathrm{m}}$ according to probability q_m
  Propose a new configuration $\mathbf{s}'$ with $\mathbf{Q}_{\mathrm{m}}(\mathbf{s}'|\mathbf{s})$
  if $\alpha_{\mathrm{m}}(\mathbf{s}, \mathbf{s}') > \mathtt{rand}[0, 1]$ **then**
    $X_{t+1} \leftarrow \mathbf{s}'$
  else
    Propose an alternative configuration $\tilde{\mathbf{s}}$ based on $\mathbf{Q}^2_{\mathrm{LT}}(\tilde{\mathbf{s}}|\mathbf{s}, \mathbf{s}')$
    if $\alpha^2_{\mathrm{LT}}(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) > \mathtt{rand}[0, 1]$ **then**
      $X_{t+1} \leftarrow \tilde{\mathbf{s}}$
    else
      $X_{t+1} \leftarrow \mathbf{s}$
    end if
  end if
  $t \leftarrow t + 1$
  Decrease the temperature: $T \leftarrow T_t$
end while

3.4 Integration of Line Hypotheses

While MPP modeling allows to design complex prior knowledge of the object distribution, its performance is very sensitive to the selection of modeling parameters and hyperparameters. For clarity, we note that the modeling parameters are related to the physical characteristics of the line segments (*e.g.,* range of length and orientation). The hyperparameters denote the weighting coefficients of energy terms (*i.e.,* ω_d, $\boldsymbol{\pi}^{\mathrm{p}}$, and ω_p). The modeling parameters can be chosen empirically since the values are related to the image resolution (see Section 3.5); however, it is hard to estimate the hyperparameters via trial-and-error for different types of dataset. Our goal is to maximize the proposed probability density without estimating hyperparameters.



(a) Input (b) $|\nabla I|$ (c) [Talbot 07] (d) [Becker 13]



(e) $\hat{\mathbf{s}}^1$ (f) $\hat{\mathbf{s}}^2$ (g) $\hat{\mathbf{s}}^3$ (h) $\mathcal{P}_{\mathbb{S}}$

Figure 3.10: Given the input image (a), we compute the gradient magnitude (b). Mathematical morphology operator, path opening [Talbot 07], is applied on such gradient magnitude image (c). Linearity score of each pixel is drawn by the supervised feature learning algorithm [Becker 13] (d). We provide line hypotheses (e)–(g) associated with different hyperparameter vectors. Composition result (h) is equivalent to mixture probability density, and it highlights pixels corresponding to linear structures.

3.4.1 Generation of K Line Hypotheses

Let $\boldsymbol{\omega} = [\omega_d \ \omega_p \ \pi_1 \ \pi_2]^\intercal$ be a hyperparameter vector which consists of the weighting coefficients of the proposed probability density. Suppose that we have K different hyperparameter vectors, $\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_k, \cdots, \boldsymbol{\omega}_K$. For each hyperparameter vector, we substitute k-th hyperparameter vector $\boldsymbol{\omega}_k$ into the proposed probability density $p(\mathbf{s}|I, \boldsymbol{\omega}_k)$. Then, we look for its optimal configuration $\hat{\mathbf{s}}^k = \text{argmax}_{\mathbf{s}\subset\Psi}\, p(\mathbf{s}|I, \boldsymbol{\omega}_k)$ via Monte Carlo sampler proposed in Section 3.3.3.

For the implementation, we bound the values of $\boldsymbol{\omega}$ due to the practical reason. More precisely, we fix ω_p as 1000 and control the other values according to the proportion of $\chi = \omega_d = 1$. We set $\boldsymbol{\omega}_1 = [\chi, 1000, -0.1\chi, -0.1\chi]^\intercal$, and then linearly increase χ to create hyperparameter vectors, *i.e.*, $\boldsymbol{\omega}_2 = [1.1\chi, 1000, -0.11\chi, -0.11\chi]^\intercal, \boldsymbol{\omega}_3 = [1.2\chi, 1000, -0.12\chi, -0.12\chi]^\intercal$, and so on. In our experiments, we use $K = 15$ to create line hypotheses.

Algorithm 3 Intergraion of K line hypotheses

1: **for all** $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K$ **do**
2: Look for an optimal set of line segments with k-th hyperparameter vector $\boldsymbol{\omega}_k$:

$$\hat{\mathbf{s}}^k = \underset{\mathbf{s}\in\Psi}{\operatorname{argmin}} \sum_{i=1}^{\#(\mathbf{s})} E_{\text{data}}(s_i) + \sum_{s_m \sim s_n} E_{\text{prior}}(s_m, s_n; \boldsymbol{\omega}_k)$$

3: Compute a probability map $\mathcal{P}_k$ via (3.29)
4: **end for**
5: Integrate K line hypotheses $\mathcal{P}_{\mathbb{S}} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{P}_k$
6: Find the most promising hyperparameter $\boldsymbol{\omega}_{\hat{k}}$:

$$\hat{k} = \underset{k=\{1,\ldots,K\}}{\operatorname{argmax}} \operatorname{CC}(\mathcal{P}_{\mathbb{S}}, \mathcal{P}_k)$$

7: Update the data likelihood energy (3.33)
8: Simulate Markov chain of the modified cost function over the reduced sample space $\mathbb{S}$ with $\boldsymbol{\omega}_{\hat{k}}$:

$$\mathbf{s}^* = \underset{\mathbf{s}\subset\mathbb{S}}{\operatorname{argmin}} \sum_{i=1}^{\#(\mathbf{s})} E'_{\text{data}}(s_i) + \sum_{s_m \sim s_n} E_{\text{prior}}(s_m, s_n; \boldsymbol{\omega}_{\hat{k}}).$$

3.4.2 Combination of Line Hypotheses into a Probability Map

We now have a family of line hypotheses $\mathbb{S} = \bigcup_{k=1}^{K} \hat{\mathbf{s}}^k$ obtained from K different hyperparameter vectors. We jointly use the image data and these line hypotheses. More specifically, the final solution $\mathbf{s}^*$ maximizes not only the probability density w.r.t. image but also the consensus among line hypotheses. For each optimal configuration $\hat{\mathbf{s}}^k$, we compute a probability map $\mathcal{P}_k$ of being a line in the image site. Then, we integrate K probability maps into a mixture density $\mathcal{P}_{\mathbb{S}}$:

$$\mathcal{P}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \exists s_i^k \in \hat{\mathbf{s}}^k, \mathbf{x} \in s_i^k, \\ \frac{1}{2} & \text{if } \exists s_i^k \in \hat{\mathbf{s}}^k, \mathbf{x} \in A(s_i^k), \\ 0 & \text{otherwise,} \end{cases} \tag{3.29}$$

$$\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{P}_k(\mathbf{x}). \tag{3.30}$$

Figure 3.10 compares image gradient magnitude, morphological filtering [Talbot 07], supervised feature learning [Becker 13], line hypotheses, and the mixture density. Since

the input image contains many high frequency components, its gradient also highlights non-linear structures in the background. While the morphological filter accentuates linear structures, its performance depends on the setting of path length. Supervised learning method requires high quality of a training dataset and corresponding ground truth images. Depending on the setting of hyperparameter vectors, the MPP model leads incomplete detection results as shown in Figure 3.10. (e)–(g). We integrate line hypotheses of the proposed MPP model into a mixture density $\mathcal{P}_{\mathbb{S}}$. The mixture density shows the consensus between line hypotheses in the sense that the pixels corresponding to line structures are more highlighted when compared to [Becker 13, Talbot 07].

We assume that the most promising hyperparameter vector draws a configuration which is more akin to the mixture density. We compute the *correlation-coefficient* (CC) between $\mathcal{P}_{\mathbb{S}}$ and $\mathcal{P}_k$'s to analyze coherence of line detection results. That is

$$\hat{k} = \underset{k=\{1,\ldots,K\}}{\operatorname{argmax}} \operatorname{CC}(\mathcal{P}_{\mathbb{S}}, \mathcal{P}_k), \tag{3.31}$$

$$\operatorname{CC}(\mathcal{P}_{\mathbb{S}}, \mathcal{P}_k) = \frac{\sum_{\mathbf{x}} \left(\mathcal{P}_{\mathbb{S}}(\mathbf{x}) - \mathbb{E}[\mathcal{P}_{\mathbb{S}}]\right)\left(\mathcal{P}_k(\mathbf{x}) - \mathbb{E}[\mathcal{P}_k]\right)}{\sqrt{\sum_{\mathbf{x}} \left(\mathcal{P}_{\mathbb{S}}(\mathbf{x}) - \mathbb{E}[\mathcal{P}_{\mathbb{S}}]\right)^2 \sum_{\mathbf{x}} \left(\mathcal{P}_k(\mathbf{x}) - \mathbb{E}[\mathcal{P}_k]\right)^2}}, \tag{3.32}$$

where $\hat{k}$ represents the index of the most reliable hyperparameter vector.

3.4.3 Curvilinear Structure Extraction from Reduced Sampling Space

The line hypotheses span a configuration space $\mathbb{S} \subset \Psi$ which will be considered as a new sample space. Since the size of $\mathbb{S}$ is significantly reduced compared to the original sample space Ψ, the optimization process becomes more tractable in terms of convergence time and detection accuracy. We redefine the data likelihood energy by adding a new energy term as follows:

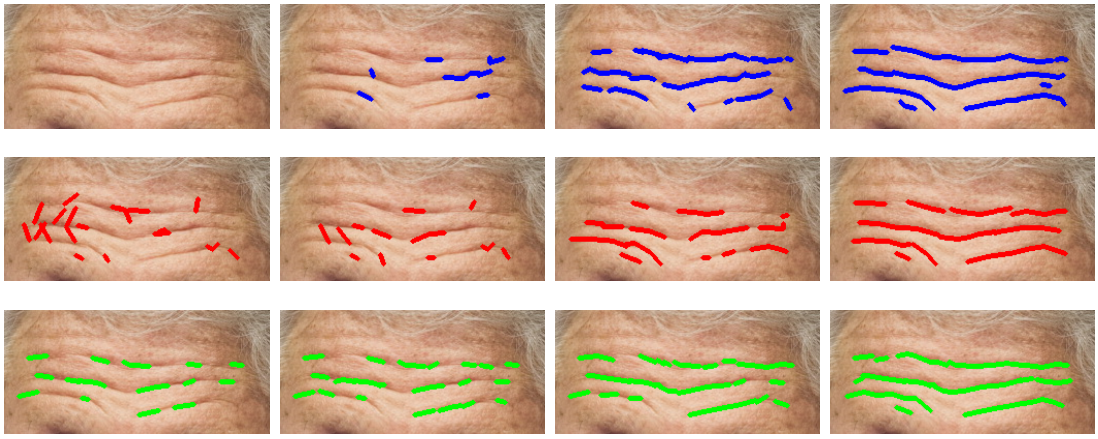
$$E'_{\text{data}}(\phi(\mathbf{x})) = E_{\text{data}}(\phi(\mathbf{x})) - \log \mathcal{P}_{\mathbb{S}}(\mathbf{x}), \tag{3.33}$$

$$E'_{\text{data}}(s_i) = \sum_{\mathbf{x}_j \in s_i} E'_{\text{data}}(\phi(\mathbf{x}_j)). \tag{3.34}$$

We simulate the modified probability density over the reduced sample space $\mathbb{S}$ with the most promising hyperparameter vector $\boldsymbol{\omega}_{\hat{k}}$:

$$\mathbf{s}^* = \underset{\mathbf{s} \subset \mathbb{S}}{\operatorname{argmax}}\, p(\mathbf{s}|I, \boldsymbol{\omega}_{\hat{k}}) \tag{3.35}$$

$$= \underset{\mathbf{s} \subset \mathbb{S}}{\operatorname{argmin}} \sum_{i=1}^{\#(\mathbf{s})} E'_{\text{data}}(s_i) + \sum_{s_m \sim s_n} E_{\text{prior}}(s_m, s_n; \boldsymbol{\omega}_{\hat{k}}). \tag{3.36}$$



(a) $T_0 = 100$ (b) $T_{10^2} = 21.67$ (c) $T_{10^3} = 14.47$ (d) $T_{10^4} = 10.86$

Figure 3.11: We provide intermediate sampling processes when the temperature parameter T_t is decreasing. The results shown in first row are obtained without specifying seed segment. For the second row, we randomly set 20 seed segments and run the algorithm. For the third row, we initialize 20 line segments which are highly corresponding to underlying curvilinear structures. The algorithm converges toward almost the same solution regardless of the initial state.

We illustrate the line hypotheses integration process in Algorithm 3.

3.5 Experiments

We test the proposed algorithm on a wide range of datasets: facial wrinkles, DNA filaments[1], road cracks, and retinas. The facial wrinkle images are collected on the Internet, and forehead areas are manually selected for the experiments. Test images of the defects on the road pavements and ground-truth are courtesy of Chambon *et al.* [Chambon 10]. We use the DRIVE dataset [Staal 04] to test the proposed algorithm on retina images.

For all test sequences, we fix the modeling parameters as follows: $\ell_{\min}$ is set to 5 pixels and $\ell_{\max} = 20$ pixels. The orientation θ is varying from $-90°$ to $90°$ with increments of $2°$. The marginal distance of connected segments ϵ is fixed to 2 pixels, and the maximum angular difference of aligned segments τ is $30°$. For the SA, the initial temperature T_0 is set to 100, and it follows the logarithm cooling schedule $T_t = {}^{T_0}/_{\log(1+t)}$, where t denotes the number of the current iteration. We start the sampling process with the empty configuration. However, careful choice of initial segments can

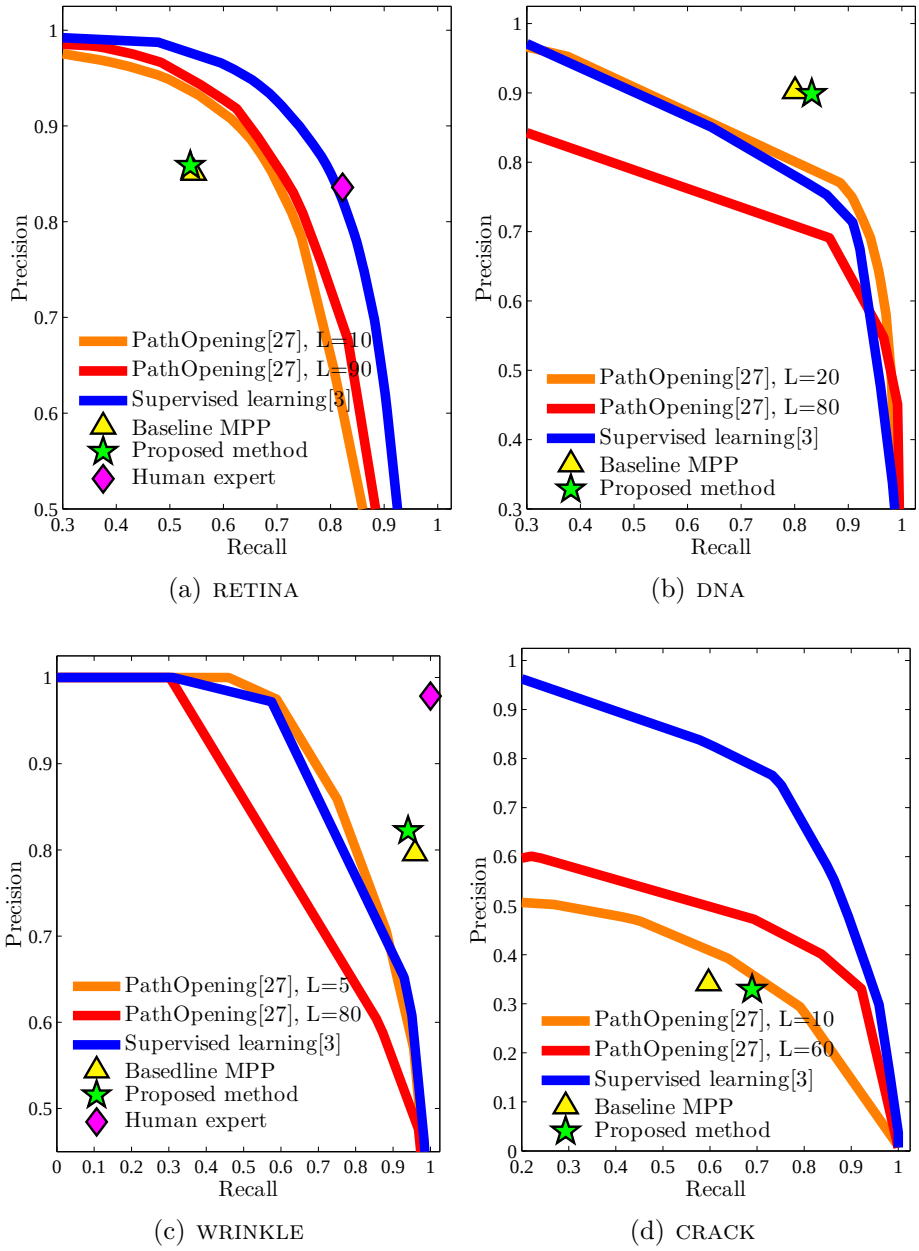
[1] https://www.biochem.wisc.edu/faculty/inman/empics/dna-prot.htm

speed up the convergence of the algorithm (see Figure 3.11). The computational time depends on the image resolution and the number of objects; it takes less than a minute for the experimental images having 300×400 pixels, approximately. We use a PC with a 2.9 GHz CPU (4 cores) and 8 GB RAM.

To compare the performances of the proposed method with the state-of-the-art techniques, we apply the path opening operator [Talbot 07] on the gradient magnitude images by controlling the length parameters. For the supervised feature learning algorithm [Becker 13], we train 15 images for each dataset. The baseline MPP is selected from the line hypotheses among which it shows the best performance. In our experiments, we use the original implementations of path opening operator[2] and supervised feature learning algorithm[3].

Figure 3.12 shows the precision-and-recall curves for four test images. To plot the precision-and-recall curve, we need to control threshold values for changing the operating point of the binary classifiers. However, the proposed stochastic algorithm generates a binary map which denotes the areas corresponding to the final line configuration. The performances of the supervised feature learning algorithm [Becker 13] are controlled by the quality of the training set; hence, it shows low performances on WRINKLE and DNA datasets, which are composed of noisy images of various sizes. In particular, the ground truth set of the WRINKLE dataset is based on subjective perception. While the morphology operator [Talbot 07] enhances linear structures on gradient magnitude images, it is required to specify the length of the linear structures according to the target applications. It might be unfair comparison with the competing algorithm using the precision-and-recall curve. One reason we have to consider is that pixelwise comparison cannot evaluate the structural similarity between ground truth map and the detected line segments. In other words, the measure can mislead the performance if the line detection results are slightly sifted compared to the groundtruth map. For example, although the proposed algorithm provides visually plausible shape reconstruction for road cracks, the quantitative performance is not the best.

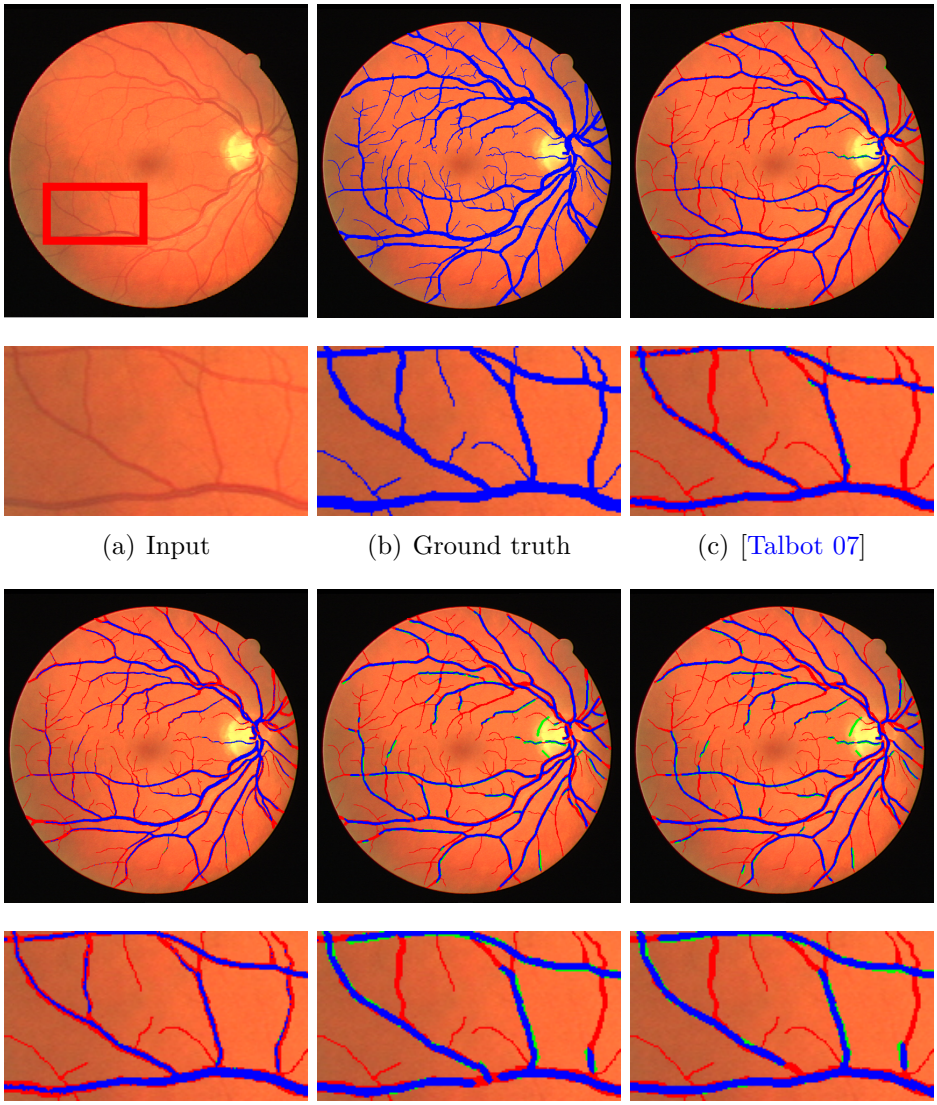
[2] http://hugues.zahlt.info/91.html
[3] http://cvlab.epfl.ch/page-108936-en.html



(a) RETINA (b) DNA

(c) WRINKLE (d) CRACK

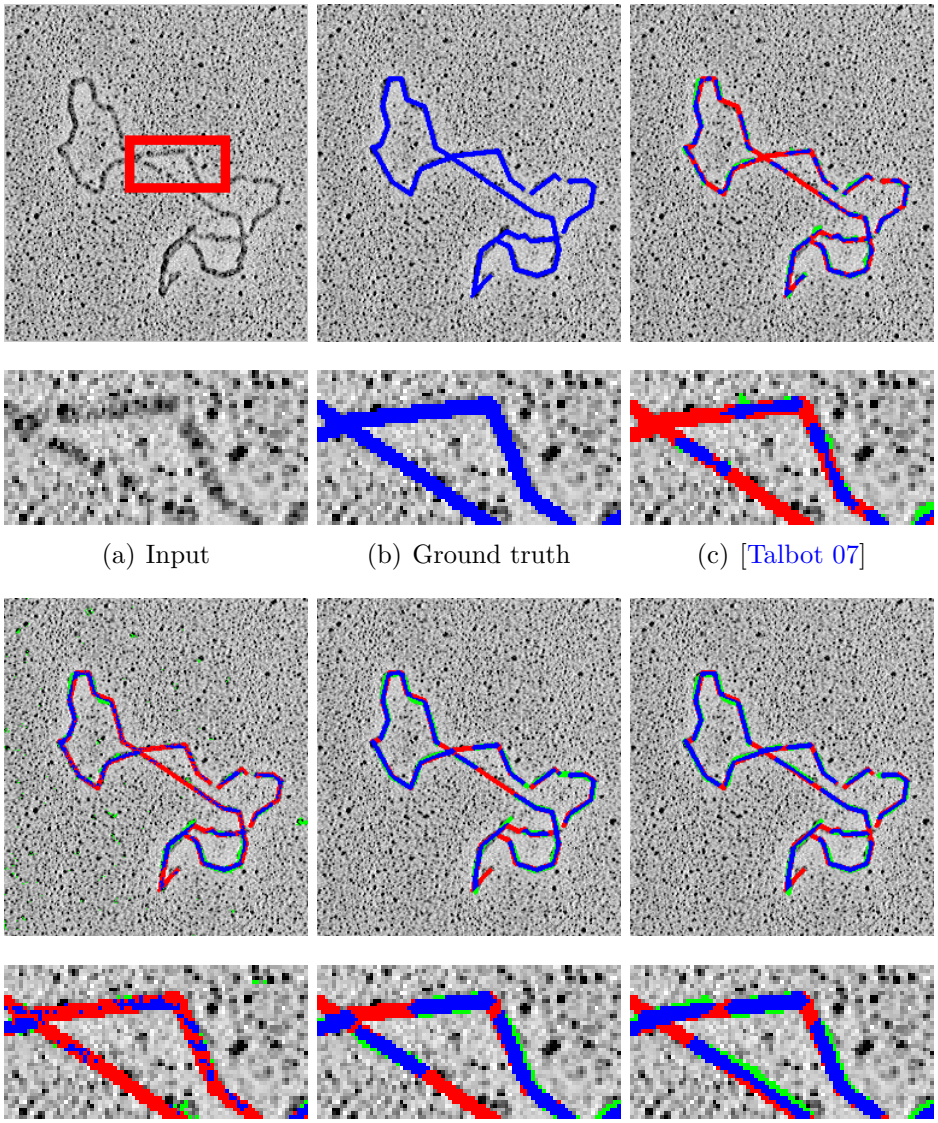
Figure 3.12: Precision-and-recall curves for pixelwise segmentation of curvilinear structures using path opening operator [Talbot 07] with different setups of length, supervised feature learning [Becker 13], baseline MPP, and the proposed method.



(a) Input (b) Ground truth (c) [Talbot 07]

(d) [Becker 13] (e) Baseline MPP (f) Proposed

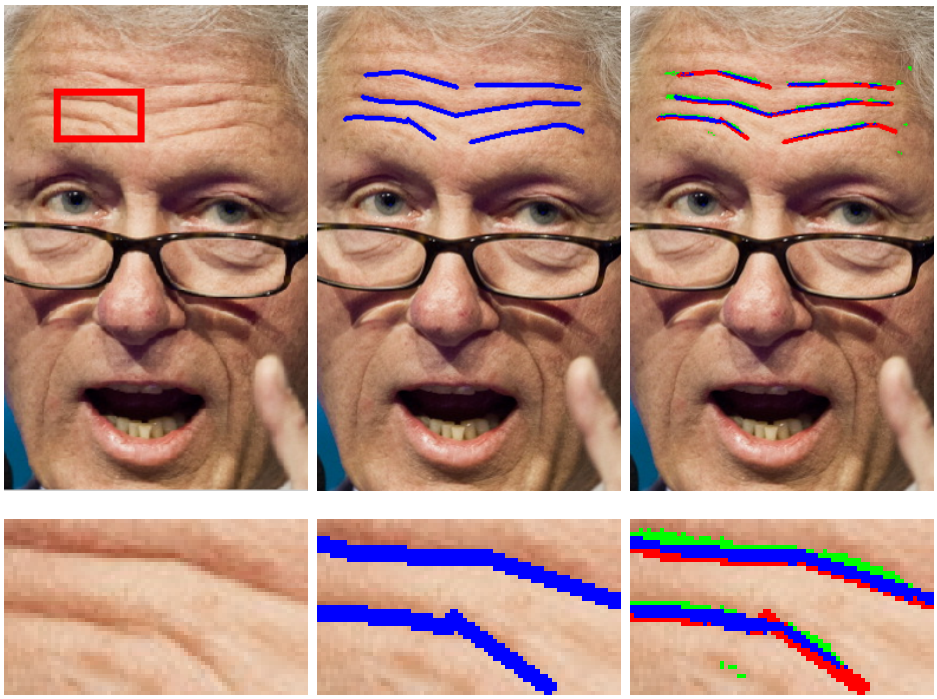
Figure 3.13: We visualize the localization of the curvilinear structures on a RETINA image. Threshold values of (c) and (d) are chosen to achieve the closest recall scores to the proposed method. We use blue pixels to indicate areas which are completely corresponding to ground truth (b). Green and red pixels denote over-detected and under-detected areas, respectively, as compared with ground truth.



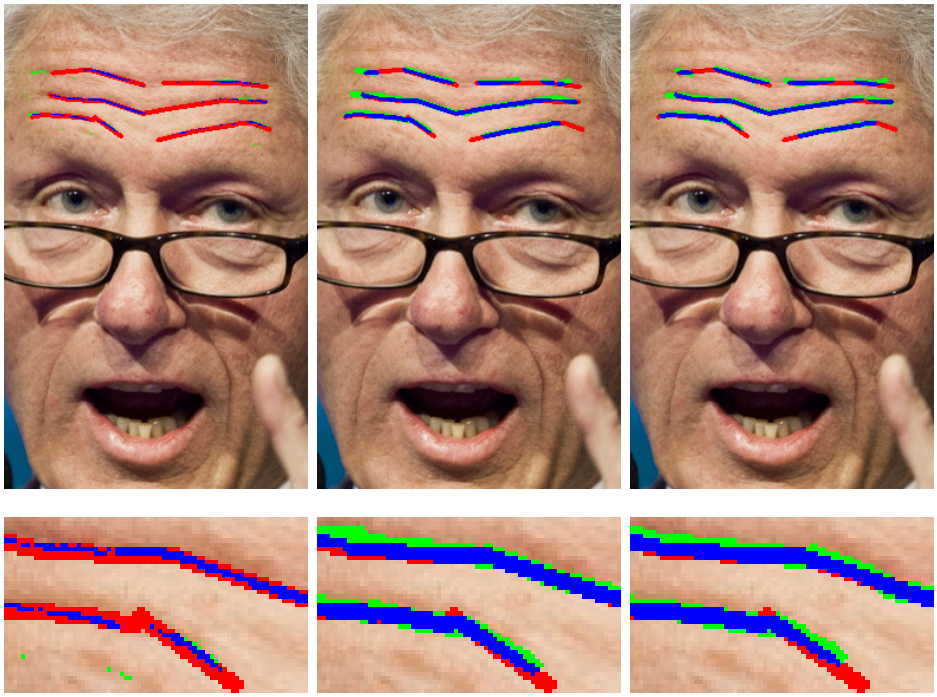
(a) Input (b) Ground truth (c) [Talbot 07]

(d) [Becker 13] (e) Baseline MPP (f) Proposed

Figure 3.14: We visualize the localization of the curvilinear structures on a DNA image. Threshold values of (c) and (d) are chosen to achieve the closest recall scores to the proposed method. We use blue pixels to indicate areas which are completely corresponding to ground turth(b). Green and red pixels denote over-detected and under-detected areas, respectively, as compared with ground truth.

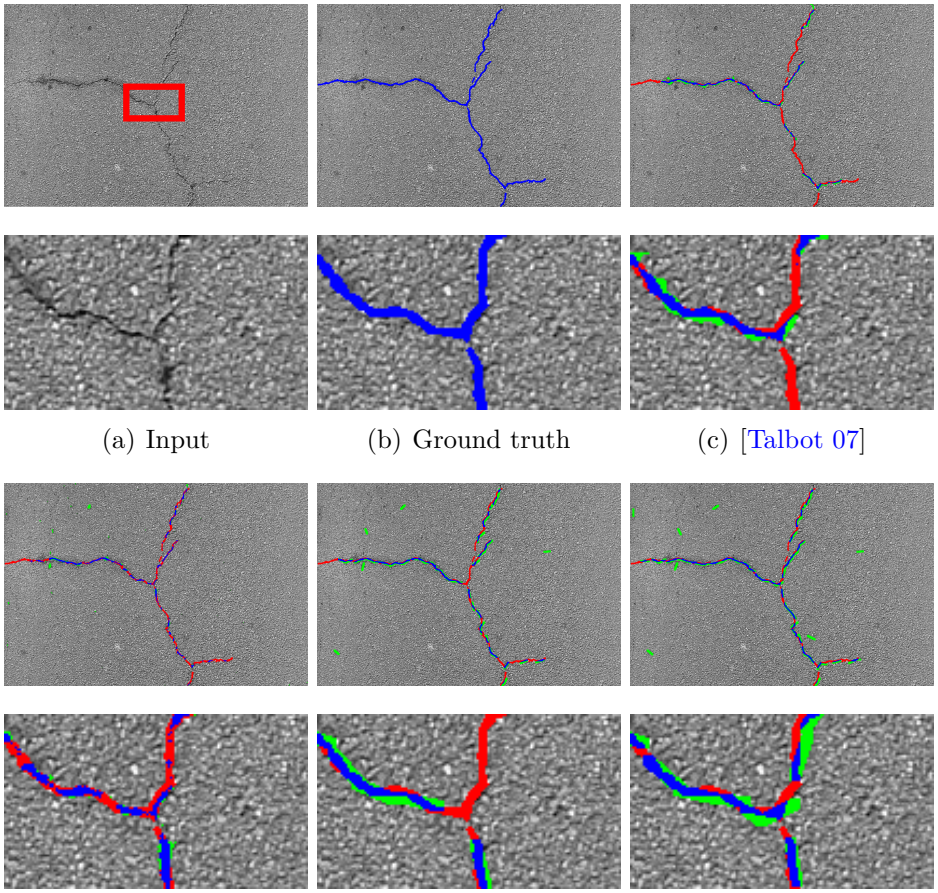


(a) Input (b) Ground truth (c) [Talbot 07]



(d) [Becker 13] (e) Baseline MPP (f) Proposed

Figure 3.15: We visualize the localization of the curvilinear structures on a WRINKLE image. Threshold values of (c) and (d) are chosen to achieve the closest recall scores to the proposed method. We use blue pixels to indicate areas which are completely corresponding to ground truth (b). Green and red pixels denote over-detected and under-detected areas, respectively, as compared with ground-truth.



(a) Input (b) Ground truth (c) [Talbot 07]

(d) [Becker 13] (e) Baseline MPP (f) Proposed

Figure 3.16: We visualize the localization of the curvilinear structures on a CRACK image. Threshold values of (c) and (d) are chosen to achieve the closest recall scores to the proposed method. We use blue pixels to indicate areas which are completely corresponding to ground truth(b). Green and red pixels denote over-detected and under-detected areas, respectively, as compared with ground truth.

We visually compare the curvilinear structure detection results of the proposed MPP model with the manually labeled image by human expert, morphology operator [Talbot 07], supervised feature learning algorithm [Becker 13], and baseline MPP. For a fair comparison, we set the threshold values of the competing algorithms [Talbot 07, Becker 13] to obtain the closest recall scores to the proposed algorithm in Figure 3.13, 3.14 3.15, and 3.16. Blue pixels denote perfectly matching regions as compared with the ground-truth. Green and red pixels show over-detected and under-detected results, respectively.

The main strength of the proposed algorithm is that it ensures stable performances for all datasets without any parameter estimation procedure. The proposed algorithm

extracts the most salient line structures in the input image. On the other hand, the proposed algorithm suffers from under-detection when the width of the line structure is varying, for example, see the result for the RETINA. Such drawback can be overcome if we introduce an additional parameter for width of the line segment in our MPP model.

3.6 Conclusions

We propose a novel MPP model to reconstruct curvilinear structures with an optimal set of line segments. For the data likelihood, the proposed density function computes rotated gradient statistics and morphological profiles. SVM reduces the sampling space according to the initially classified areas and defines statistics of the curvilinear features. Prior distribution of the proposed MPP model define interactions of the local configuration in terms of coupling energy states and overlapping areas. We have presented a new optimization scheme which is not biased by the parameter selection in the MPP model. We used an RJMCMC sampler with delayed rejection kernel for different hyperparameter vectors to obtain multiple line hypotheses. The line hypotheses span a feasible sample space, so that the final solution interprets underlying curvilinear structures more faithfully. We have shown line detection results on a wide rage of datasets, and compared the performances of the proposed method with morphological filtering [Talbot 07], supervised learning [Becker 13], and baseline MPP method. The whole optimization process is friendly designed with respect to a parallel implementation; thus, the computational time can be further reduced by applying the parallel Monte Carlo sampler [Verdié 14].

Chapter4

Chapter 4

Inference of Curvilinear Structure based on Learning a Ranking Function and Graph Theory

Abstract

To detect curvilinear structures in natural images, we propose a novel ranking learning system and an abstract curvilinear shape inference algorithm based on graph theory. We analyze the curvilinear structures as a set of small line segments. In this chapter, the rankings of the line segments are exploited to systematize the topological feature of the curvilinear structures. Structured Support Vector Machine is employed to learn the ranking function that predicts the correspondence of the given line segments and the latent curvilinear structures. We employ image gradient profiles and morphological profiles to describe curvilinear features. Also, we propose an orientation-aware feature descriptor and a feature grouping operator to improve the structural integrity during the learning process. To infer the curvilinear structure, we build a graph using the output rankings of the line segments. We progressively reconstruct the curvilinear structure by looking for paths between remote vertices in the graph. Experimental results show that the proposed algorithm faithfully detects the curvilinear structures within various datasets.

Contents

ASSUME that the entire curvilinear structure can be decomposed into many straight line segments. We learn a ranking function that predicts the shape similarity between individual line segments and the latent curvilinear structure. We propose an orientation-aware curvilinear feature descriptor for the learning system, which consists of the image gradient profiles and morphological profiles. Structured Support Vector Machine (SSVM) [Tsochantaridis 05] is employed to obtain weight coefficients for the proposed feature descriptor using the training dataset. To infer the structural information, we build an undirected and weighted graph based on the output rankings of the line segments. We then reconstruct the coarse curvilinear structure by exploring paths of vertices maximizing their distance in the graph. Unlike the previous approaches such as segmentation of the curvilinear structure [Becker 13] and centerline detection [Sironi 14], the proposed algorithm can provide the topological importance level of the curvilinear structure. Figure 4.1 compares results of curvilinear structure segmentation [Becker 13], centerlines detection [Sironi 14], and the proposed method.

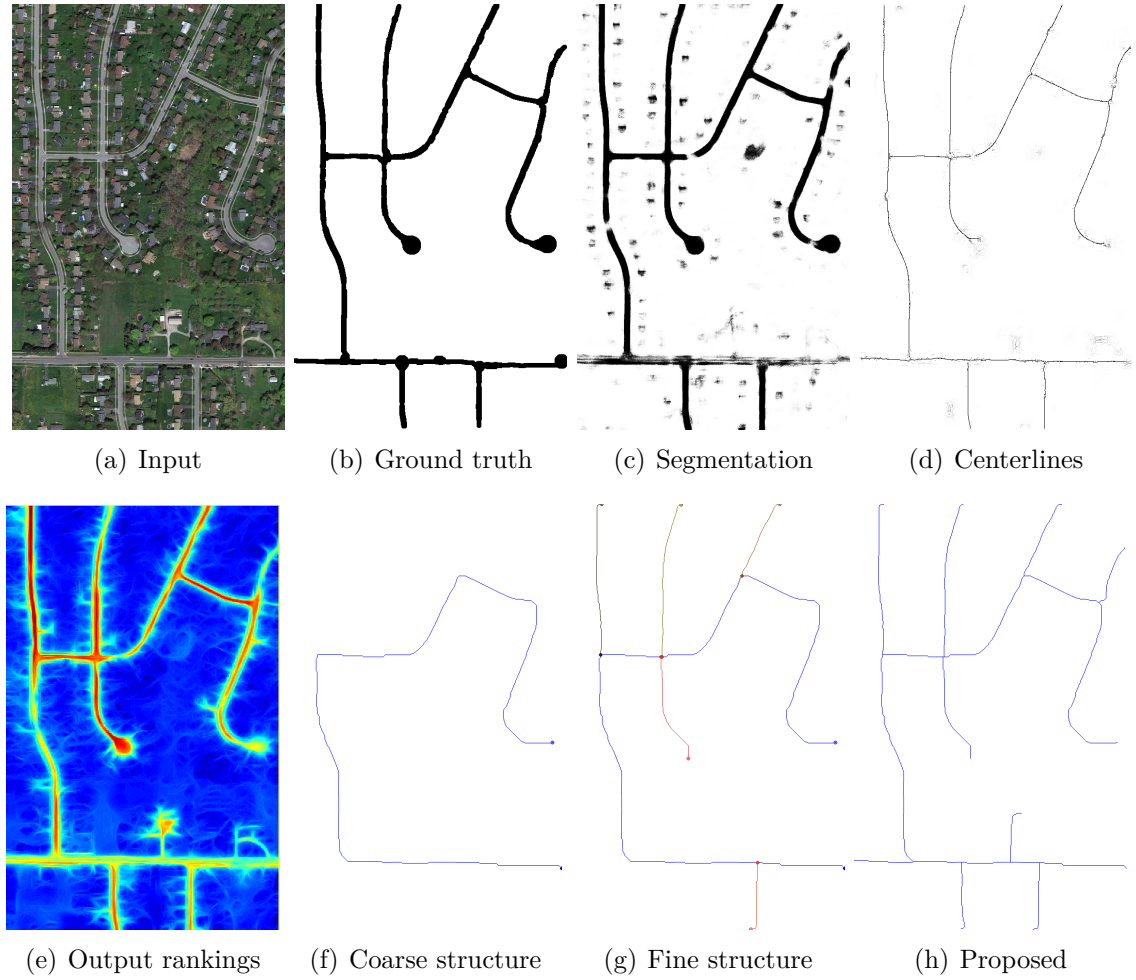
To apply a learning system, we need an accurately annotated ground truth for training datasets, where the ground truth is manually labeled by a human expert to classify pixels on the curvilinear structure. It is difficult to secure enough quantity of ground truth images with high quality in that the manual segmentation of curvilinear structure is time consuming work with great efforts. To address this problem, we evaluate a line segment using the shape similarity measure whether it belongs to the latent curvilinear structure. We minimize corruptions of the learning process due to the erroneous annotations.

We also study the graph theory for shape simplification of the complex curvilinear structure. Although the centerline is able to encode the scale of the curvilinear structure, it is insufficient to interpret the topological level of the curvilinear structure. The structured output rankings of the line segment are used to draw the latent curvilinear structure in the order of topological importance. Thus, we create a weighted graph to reconstruct the curvilinear structure with consideration for such topological order. We search for the longest path which minimizes the cumulative rankings of the line segments on the path.

The main contributions described in this chapter are summarized as follows:

- We propose an orientation-aware curvilinear feature descriptor for the ranking learning system;
- We learn a ranking function to predict the correspondence between line segments and curvilinear structure;
- We reconstruct the curvilinear structure with topological order based on graph theory; and

- We test and validate the proposed algorithm on numerous datasets containing micro and macro scales of curvilinear structures in the nature.



(a) Input (b) Ground truth (c) Segmentation (d) Centerlines

(e) Output rankings (f) Coarse structure (g) Fine structure (h) Proposed

Figure 4.1: Comparing with the segmentation ((c), [Becker 13]) and the centerline detection ((d), [Sironi 14]) methods, the proposed algorithm is able to organize the topological importance of the curvilinear structure. The output rankings (e) highlight areas according to the geometric similarity of line segments and the underlying curvilinear structure. Our graphical model represents different levels of detail of the latent curvilinear structure with the minimum number of pixels. The proposed algorithm reconstructs the coarsest structure (f), and then iteratively adds fine branches (g). The final result (h) is obtained by a few iterations.

Das *et al.* [Das 11] employs SVM to train features and applies skeletonization technique. The overall procedure to detect curvilinear structure is similar to the proposed algorithm. In detail, the proposed algorithm aims to reconstruct curvilinear structures

which are latent in different types of image. Thus, we exploit curvilinear features which can encode structural attributes, *i.e.,* image gradient and morphological features. On the other hands, Das *et al.*'s method targets on a specific application (satellite imaging) so that its feature vectors mainly consist of photometric cues such as RGB histogram features. Use of photometric cue is weak for detecting curvilinear structures with noisy background textures. Also, the proposed progressive path reconstruction algorithm can visualize the topological features in different levels of detail. However, medial axis transform (MAT) employed in [Das 11] requires an additional process to prune the redundant outcomes which are not associated with the latent curvilinear structure. Unsalan and Sirmacek [Unsalan 12] proposed a road network detection algorithm based on locally predicted probability density function. Thus, the algorithm consists of several refinement steps to obtain the final results, where each step requires manual setting of threshold probability. Unsalan and Sirmacek's method is limited to test many different types of images containing curvilinear structure. On the other hands, the proposed model finds globally optimized line configuration and is capable to detect curvilinear structures on retinal images, biological images, aerial images, and asphalt surfaces.

The rest of the chapter is organized as follows: Section 4.1 reviews the machine learning algorithm on which the proposed method based. Section 4.2 provides the outline of our method. Section 4.3 proposes an orientation aware curvilinear feature descriptor. Section 4.4 explains how we learn the ranking function. Section 4.5 develops a graphical model for inference of curvlinear structure. Section 4.6 shows experimental results on different types of datasets. Finally, Section 4.7 concludes this chapter.

4.1 Revisited machine learning systems

We want to extract the structured information of the line networks that appear within homogeneous background textures. Since the information embedded in a single pixel is limited to infer the latent structure, Markov Random Fields (MRF, [Geman 84, Geman 90, Besag 86, Sziranyi 14]) or Conditional Random Fields (CRF, [Lafferty 01, Kumar 06]) based models have been developed to enforce the label consistency on the pre-organized output space. *e.g.,* neighborhood pixels are assumed to have the same label with the high probability when minimizing the cost function. However, the topology of the curvilinear structure composed of line objects is too intricate to be applicable with such design approaches. It is also impossible to specify the topology of the complex line networks with a few parameters. Instead, we employ a machine learning framework, SSVM proposed by [Tsochantaridis 05], to obtain undiscovered distribution which contains the structured relation of input-and-output pairs. The SSVM is often comparable with the graphical models which contain the pairwise struc-

ture. Therefore, the SSVM framework is favored to solve computer vision problems, such as object segmentation [Bertelli 11, Lucchi 12, Kim 14], pose estimation algorithm [Yang 11], and multi-class classification [Mittal 12]. In this work, we use the SSVM framework to train a ranking function that predicts the compatibility with tokenized line segments and the latent curvilinear structures. Finally, the structured output rankings of the line segments are used to estimate the local orientation and to reconstruct the latent curvilinear structures as an ensemble of high ranked line segments. In the remainder of this section, we review Support Vector Machines (SVM) [Vapnik 95] and structured SVM on which the proposed algorithm is based.

4.1.1 Support Vector Machines

The goal of machine learning system is to obtain a function $h : \mathcal{X} \mapsto \mathcal{Z}$ which maps a feature space $\mathbf{x} \in \mathcal{X}$ to an output space $z \in \mathcal{Z}$. Using a given training dataset $\mathcal{D} = \{(\mathbf{x}_i, z_i)\}_{i=1}^{K}$ consisting of input and output pairs, a supervised machine learning algorithm evaluates the quality of a hypothesis $\hat{z} = h(\mathbf{x})$. Consider a classification problem, we want to find an optimal hypothesis that divides data points into disjoint areas. Figure 4.2 provides a toy example to compare the quality of a hypothesis which is generated by a linear separator denote by black thick line in this figure. Intuitively, we prefer to employ the separator in the bottom row due to the input noise tolerance. Let add a new input point (denoting x in the second columns of the figure) to test and see the behaviors of the two hypotheses. While the input point is spatially closer to the group of red circles than that of blue triangles, the hypothesis of the first row classifies the point as the group of blue triangles. Furthermore, the input point can be perturbed by noise, so that the decision is less reliable at the region near to the boundary. In other words, the noise tolerance can be measured by the distance between a data point and the decision boundary. Thus, the maximum noise tolerance (margin) of the hypothesis is computed using a data point which is the closest to the decision boundary. This is the key idea of the SVM classifier [Vapnik 95, Joachims 99] which maximizes margin for separating given data points.

Now, we develop a binary SVM classifier in detail. Suppose that we have K training data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \subset \mathbb{R}^d$ and the corresponding labels are $z_i \in \{-1, 1\}$, then a hypothesis is defined as $h : \mathbb{R}^d \mapsto \{-1, 1\}$, where d denotes the dimension of the input feature space. Algebraically, a hypothesis is specified as a linearly separable hyperplane $\mathbf{w}^\intercal \mathbf{x} + b = 0$, where $\mathbf{w}$ and b are the normal vector and the bias, respectively. So, for classification, we simply check the sign of a given input data point $\mathbf{x}$ with respect to the hyperplane:

$$h(\mathbf{x}) = \mathtt{sign}(\mathbf{w}^\intercal \mathbf{x} + b) \tag{4.1}$$

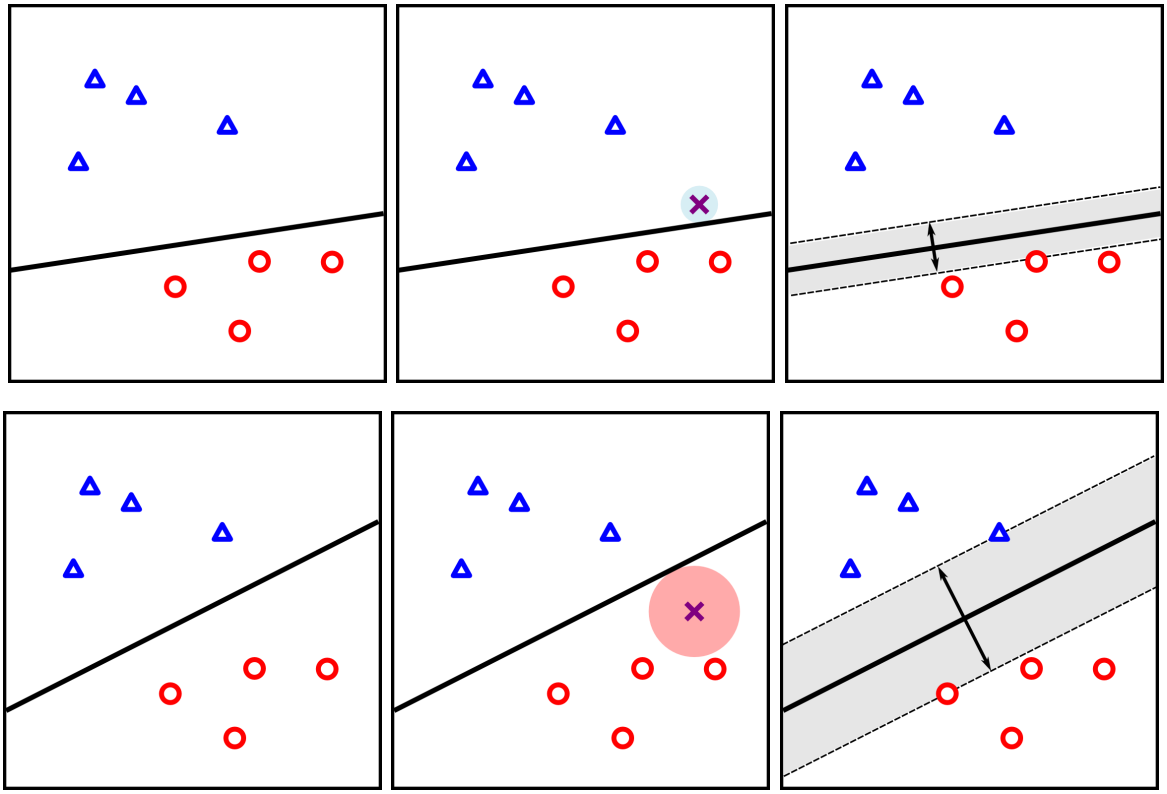


Figure 4.2: For the same data points, we compare two linear separators. Intuitively, the separator in the bottom row is safer in that its decision boundary can tolerate a large noise. The maximum noise tolerance of a separator is called its margin corresponding to the thickness of the gray areas in this figure. Thus, SVM is designed to find a hypothesis with the maximum margin for the given training data points.

Let $\mathbf{x}'$ be a point on the hyperplane. Geometrically, the distance from an arbitrary point $\mathbf{x}$ to the hyperplane h is computed by the projection of a vector $(\mathbf{x}-\mathbf{x}')$ onto $\mathbf{w}$, which is given by

$$\texttt{dist}(\mathbf{x}, h) = \frac{|\mathbf{w}^\intercal \mathbf{x} - \mathbf{w}^\intercal \mathbf{x}'|}{\|\mathbf{w}\|} = \frac{|\mathbf{w}^\intercal \mathbf{x} + b|}{\|\mathbf{w}\|}, \tag{4.2}$$

where $\mathbf{w}^\intercal \mathbf{x}' + b = 0$. We assume that $z_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1$ to separate all data points with enough margin. It is also called *hard margin* constraint. Using the fact that $z_i(\mathbf{w}^\intercal \mathbf{x}_i + b) > 0$, the distance of a training data point $\mathbf{x}_i$ to h can be written as

$$\texttt{dist}(\mathbf{x}_i, h) = \frac{z_i(\mathbf{w}^\intercal \mathbf{x}_i + b)}{\|\mathbf{w}\|}. \tag{4.3}$$

As we discussed in the beginning of this section, the maximum margin is associated

with the distance of the nearest data point to the hyperplane. That is

$$\min_{i=1,\ldots,K} \mathtt{dist}(\mathbf{x}_i, h) = \frac{\min\limits_{i=1,\ldots,K} z_i(\mathbf{w}^\intercal \mathbf{x}_i + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}. \tag{4.4}$$

We then formulate the following constrained optimization problem to learn an optimal hyperplane which separates binary classes with the maximum margin:

$$\begin{array}{ll} \min\limits_{b,\mathbf{w}} & \frac{1}{2}\mathbf{w}^\intercal \mathbf{w} \\ \text{subject to} & z_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, \ldots, K \end{array} \tag{4.5}$$

The points which are in the uncertain areas to classify contribute to maximize the margin. In this sense, we conclude that *support vectors* $\mathbf{w}$ are obtained by the nearest points to the hyperplane. The solution of (4.5) is feasible for a duality of a convex optimization problem [Boyd 04]. More precisely, we solve the following convex optimization problem with Lagrangian multipliers α_i's:

$$\mathcal{L}(b, \mathbf{w}, \alpha) = \frac{1}{2}\mathbf{w}^\intercal \mathbf{w} + \sum_{i=1}^{K} \alpha_i (1 - z_i(\mathbf{w}^\intercal \mathbf{x}_i + b)). \tag{4.6}$$

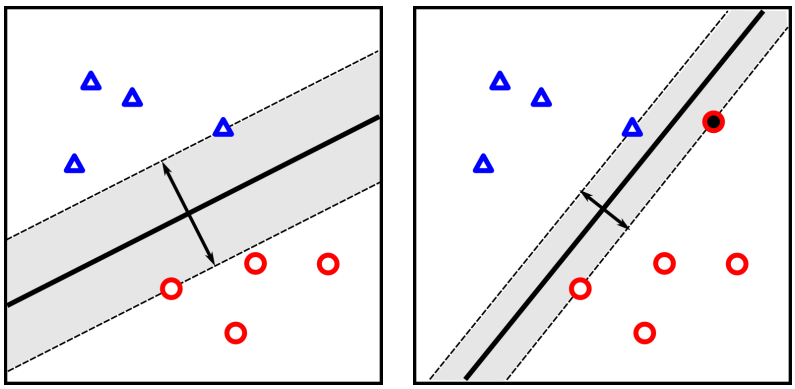
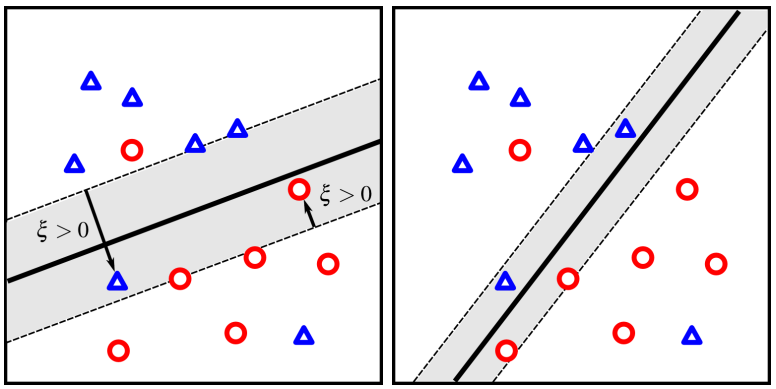


Figure 4.3: Even a single outlier (see black dot) can dramatically change the characteristic of the hyperplane.

While the hard margin constraint is a good strategy for linearly separable dataset, it also induces the over-fitting problem. Moreover, if the given training dataset includes an outlier, decision bound can be dramatically changed (see Figure 4.3). Note that, in many cases, real data are not linearly separable so that the hard margin constraint makes the hypothesis of SVM unpredictable. Thus, we introduce slack variables $\xi_i \geq 0$



(a) small C (b) large C

Figure 4.4: For the given linearly non-separable data points, we employ *soft margin* SVM which allows small classification errors around margin. A slack variable measures the such violation on the marginal areas. We modify the objective function of SVM to take into account the amount of margin violation with a parameter C. The hyperplane on the left figure is obtained using a small C so that it is less sensitive for classification errors and margin violations. On the other hand, the hyperplane on the right figure is more strict with respect to the margin violation by employing a larger C value.

to consider some violated data points which are on the marginal areas (see Figure 4.4). We modify the objective function of SVM (4.5) to take into account the amount of margin violations as the sum of slack variables such as:

$$\begin{aligned} \min_{b,\mathbf{w},\xi} \quad & \tfrac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \tfrac{C}{K}\textstyle\sum_{i=1}^{K}\xi_i \\ \text{subject to} \quad & z_i(\mathbf{w}^\intercal\mathbf{x}_i + b) \geq 1 - \xi_i; \quad \text{for } i = 1, \ldots, K \end{aligned} \tag{4.7}$$

where C is a parameter to control the relative importance of total sum of margin violation. We expect that the second term should be minimized to obtain the smallest classification errors. Thus, a large C value induces a hyperplane with thin marginal areas. This modified model is also called *soft margin* SVM which permits some points to invade the marginal areas.

4.1.2 Structured Support Vector Machines

Despite of the success of SVM models for classification, it is hard to interpret the meaning of the obtained SVM model parameters except that the support vectors are geographically proposed to separate the given training data with the maximum margin. Furthermore, to solve multi-class classification problem with binary SVM classifier, we have to assume the statistical independence between features. More precisely, we isolate

one class from the others and then iteratively train SVM classifier to obtain hypothesis of each class. In this scenario, we fails to exploit the correlation information which is latent in the data.

Researchers have developed structure prediction models to deal with complex and interdependent features [Taskar 03, Tsochantaridis 04, Joachims 09]. The joint feature of input-and-output pairs $\Psi(\mathbf{x}, z) : \mathcal{X} \times \mathcal{Z} \mapsto \mathbb{R}^N$ is used to learn structured information. It generates a vector by applying structure z to the given input feature $\mathbf{x}$. For instance, to learn the language structure (grammar), $\mathbf{x}$ and z correspond to sample sentence and grammatical structure, respectively. For a ranking learning system, $\mathbf{x}$ refers to a list of documents and z is the associated rankings of the documents in the list. A scoring function $H : \mathcal{X} \times \mathcal{Z} \mapsto \mathbb{R}$ is defined as the inner product of a model parameter $\mathbf{w} \in \mathbb{R}^N$ and the joint feature $\Psi(\mathbf{x}, z)$, which measures compatibility of the input feature and predicted structure:

$$H(\mathbf{x}, z; \mathbf{w}) = \mathbf{w}^\intercal \Psi(\mathbf{x}, z). \tag{4.8}$$

Then, we predict output structure for given model parameter $\mathbf{w}$, which maximizes the scoring function:

$$\hat{z} = \underset{z \in \mathcal{Z}}{\operatorname{argmax}}\, H(\mathbf{x}, z; \mathbf{w}). \tag{4.9}$$

For this purpose, we assume that a score with correct structure should be higher than one with incorrect structure. We formulate an optimization problem to predict underlying structure as follows:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \tfrac{1}{2}\mathbf{w}^\intercal \mathbf{w} + \tfrac{C}{K}\textstyle\sum_{i=1}^{K} \xi_i \\ \text{subject to} \quad & \mathbf{w}^\intercal \Psi(\mathbf{x}_i, z_i) - \mathbf{w}^\intercal \Psi(\mathbf{x}_i, \hat{z}_i) \geq \Delta(z_i, \hat{z}_i) - \xi_i; \\ & \forall \hat{z}_i, \hat{z}_i \neq z_i, i = 1, \ldots, K, \end{aligned} \tag{4.10}$$

where $\Delta(z_i, \hat{z}_i)$ denotes a loss function which computes the amount of the loss due to the prediction of $\hat{z}_i$ comparing to the ground truth z_i. Therefore, for all training data, we constrain that difference between the scores computed by correct structure z_i and incorrect structure $\hat{z}_i$ is always greater than a required margin. Due to the number of constraints, we need an efficient algorithm to solve the given optimization problem (4.10). Plane-cutting algorithm [Joachims 09] iteratively adds constraints until the prediction error reaches within a pre-defined tolerance value ϵ. The algorithm begins to solve (4.10) without constraints, and finds the most violated constraints against for each sample data and the corresponding prediction. For constraints which violate the required margin, plane-cutting algorithm adds $\hat{z}$ into a working set $\mathcal{W}_i$ and then solve (4.10) with respect to the updated constraints. This process is also summarized in Algorithm 4.

Algorithm 4 Cutting plane algorithm [Joachims 09]

Inputs:
  $\mathcal{D} = \{(\mathbf{x}_i, z_i)\}_{i=1}^{K} \sim$ a training dataset consisting of input-and-output pairs
  $C \sim$ controls a relative importance of regularization and regression terms
  $\epsilon \sim$ a tolerance
Output:
  $\mathbf{w} \sim$ an optimal model parameter
$\mathcal{W}_i \leftarrow \emptyset$ for all $i = 1, \ldots, K$
while no $\mathcal{W}_i$ has changed during iteration **do**
  for $i = 1, \ldots, n$ **do**
    $H(\mathbf{x}, z; \mathbf{w}) = \Delta(z_i, z) + \mathbf{w}^\intercal \Psi(\mathbf{x}_i, z) - \mathbf{w}^\intercal \Psi(\mathbf{x}_i, z_i)$
    Compute $\hat{y} = \mathrm{argmax}_{z \in \mathcal{Z}} H(\mathbf{x}, z; \mathbf{w})$
    Compute $\xi_i = \max\{0, \max_{z \in \mathcal{W}_i} H(\mathbf{x}, z; \mathbf{w})\}$
    if $H(\mathbf{x}, \hat{z}; \mathbf{w}) > \xi_i + \epsilon$ **then**
      $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\hat{z}\}$
      $\mathbf{w} \leftarrow$ optimize over $\mathcal{W} = \bigcup_i \mathcal{W}_i$
    end if
  end for
end while

4.2 Overview of the proposed algorithm

In this section, we define notations to represent curvilinear structures, and provide an overview of the proposed algorithm. Assume that an image I contains a curvilinear structure. We denote the latent curvilinear structure $g : I \mapsto \{0, 1\}$ for any pixel $\mathbf{x}$:

$$g(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is on the curvilinear structure,} \\ 1 & \text{background.} \end{cases} \tag{4.11}$$

This function is also employed to create a ground truth map, which is manually labeled, for the machine learning system and the performance evaluations.

Instead of classifying each pixel whether it belongs to the curvilinear structure, we gather line segments that agree with the latent curvilinear structure of the given image. Hence, the proposed algorithm aims to find a set of line segments $\mathbf{s} \subseteq \mathcal{S}$ highly corresponding to the curvilinear structure. We define a line segment $s \in \mathcal{S}$ as a tuple of the center coordinate $\mathbf{x}$, fixed length ℓ, fixed thickness τ, and quantized orientations θ. That is $s = (\mathbf{x}, \ell, \tau, \theta) \in I \times \mathbb{R}^2 \times \Theta$, where Θ is set of elements in $[0,\ 2\pi[$.

To train a ranking function with SSVM, we propose a novel curvilinear feature descriptor $\phi : \mathcal{S} \mapsto \mathbb{R}^N$ that assigns a feature vector $\phi(s) \in \mathbb{R}^N$ to each line segment s, where N denotes the dimension of the feature space. Specifically, the feature de-

scriptor consists of the morphological profiles ϕ_{M} to detect the expected length of the structure and the steerable filtering responses ϕ_{S} to accentuate directional properties (Section 4.3.1). To take into account the directional information of each line segment, we propose an orientation-aware feature vector $\phi'(s)$ that permutes the elements in the steerable feature vector of $\phi_{\mathrm{S}}(s)$ according to orientation θ of the given line segment s (Section 4.3.2). Also, we design a spatial neighborhood system for the line segments to improve the consistency of output ranking values (Section 4.3.3). A model parameter $\mathbf{w} \in \mathbb{R}^N$ is obtained via machine learning system (SSVM), which determines the relative importance of elements in the curvilinear feature vectors. The output rankings of line segments $z \in \mathcal{Z}$ are associated to the plausibility for each line segment s to belong to the ground truth map g. For example, a higher ranking is assigned to the given line segment if it fits well to the latent curvilinear structure (Section 4.4). Figure 4.5 summarizes the process how we learn a model parameter of the proposed curvilinear feature descriptor with the given training dataset.

With the rankings of the line segments, we build an undirected and weighted graph $G = (V, E)$ to infer the curvilinear structure with a set of elementary paths on the graph G. Finally, we iteratively reconstruct the curvilinear structure by collecting paths between remote vertices in the graph (Section 4.5).

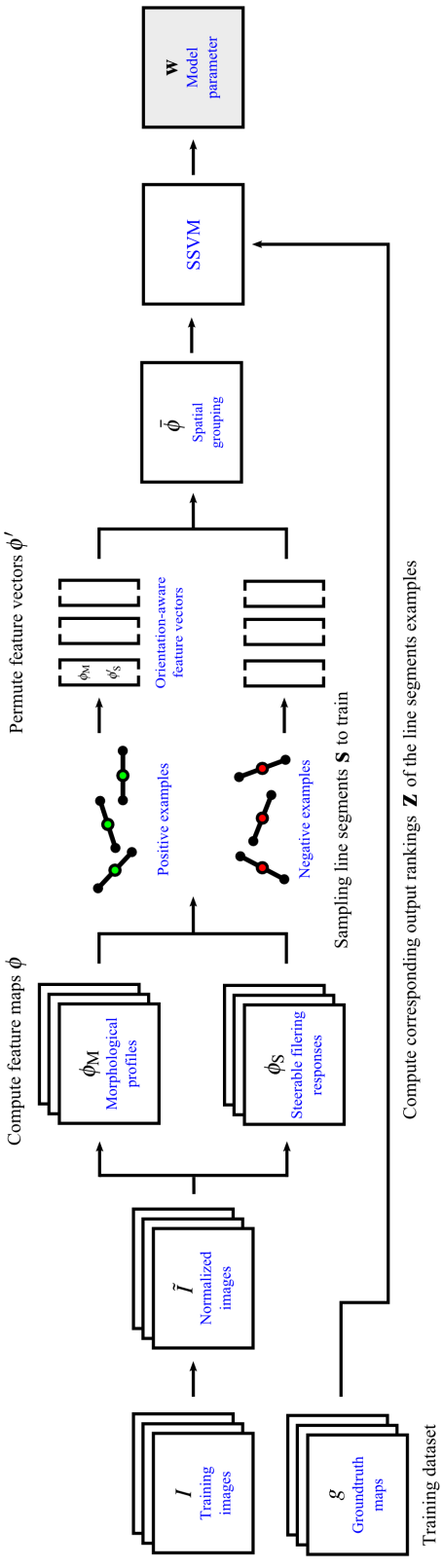


Figure 4.5: Overview of the curvilinear feature descriptor generation (Section 4.3) and ranking learning system (Section 4.4)

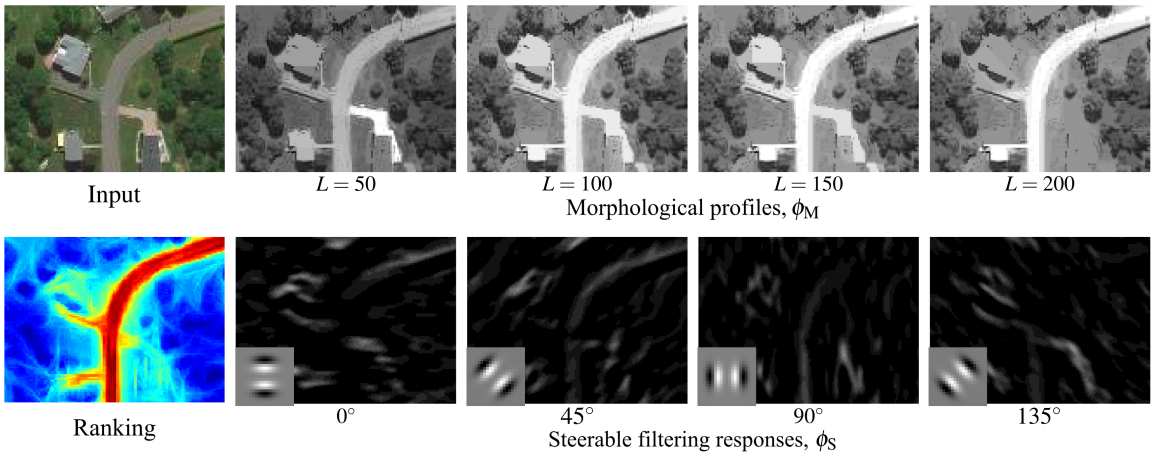


Figure 4.6: Example of the feature maps of the morphological profile ϕ_M with different length parameters $L \in \{50, 100, 150, 200\}$ and those of the steerable filtering responses ϕ_S with variable orientations $\theta \in \{0°, 45°, 90°, 135°\}$. The model parameters, which define the relative weights of those feature maps, are obtained via the ranking learning system. The higher ranking scores (denoted by red color) correspond to the latent curvilinear structure.

4.3 Orientation-Aware Curvilinear Feature

4.3.1 Feature extraction

This section is devoted to compute the curvilinear feature descriptors that are used for the inputs of the ranking learning system. The evidence of being the curvilinear structure is measured by the homogeneity of the pixel intensities and the directional responses on the path. We extract curvilinear features using the morphological profiles $\phi_M : \mathcal{S} \mapsto \mathbb{R}^4$ and the steerable filter responses $\phi_S : \mathcal{S} \mapsto \mathbb{R}^{24}$. Figure 4.6 depicts an example of the curvilinear feature maps. We refer to Chapter 2 for implementation details of the morphological operation and the steerable filters.

Before applying the filtering operations, we normalize the training and test images to remove the effects of various illumination factors:

$$\tilde{I} = \frac{1}{1 + e^{-\frac{I-\beta}{\alpha}}}, \tag{4.12}$$

where $\alpha = \max(I) - \min(I)$ represents the range of intensity values, and $\beta = \mathbb{E}[I]$ is the sample mean of the image.

To obtain the morphological profile, we apply the *path opening* operator [Talbot 07]

with different length parameters L. The path opening operator brightens pixels belonging to the linear shaped objects while it removes other types of structural elements. As the length parameter of the path opening operator is increased, we obtain a map which emphasizes longer sequence of pixels in the image.

The steerable filtering responses are associated with the directional characteristics of the given line segment. Let $\mathbf{F}_\theta$ be a steerable filter that accentuates orientation θ. The steerable feature map is computed by the convolution of $\mathbf{F}_\theta$ and the normalized image $\bar{I}$. We construct the baseline steerable filtering feature vector by stacking such steerable filtering responses with increasing order of orientation. That is

$$\phi_S(\mathbf{x}) = \left[\left(\mathbf{F}_{\theta_1} * \tilde{I}\right)(\mathbf{x}) \;\; \left(\mathbf{F}_{\theta_2} * \tilde{I}\right)(\mathbf{x}) \;\; \cdots \;\; \left(\mathbf{F}_{\theta_k} * \tilde{I}\right)(\mathbf{x}) \right]^\mathsf{T}, \tag{4.13}$$

where $\theta_1 < \theta_2 < \cdots < \theta_k$ and $k = |\Theta|$ is the number of quantized orientations in $\Theta \subset \{0, 2\pi\}$.

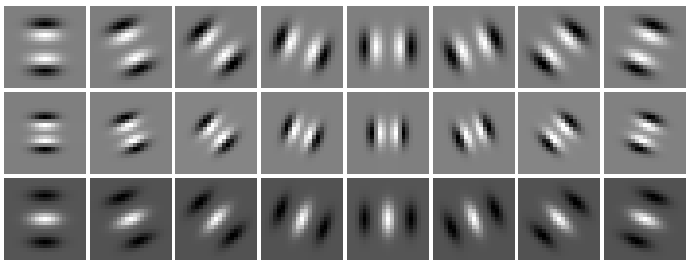


Figure 4.7: Steerable filter banks $\mathbf{f}_\theta$ to compute curvilinear feature maps

In this work, we use 4 different length parameters to create the morphological profiles and 24 filter banks to build the steerable feature maps. Specifically, the steerable filter banks consist of 3 basis kernels with respect to 8 different orientations. Figure 4.7 shows the steerable filter banks which are employed in this work.

Finally, the feature vector of the given line segment $\phi(s)$ is generated by the concatenation of the feature vectors of $\phi_\mathrm{M}(s)$ and $\phi_\mathrm{S}(s)$. We accumulate the values of the feature maps on the line segment feature and rescale those values according to length of the line segment ℓ:

$$\phi(s) = \begin{bmatrix} \phi_\mathrm{M}(s) \\ \phi_\mathrm{S}(s) \end{bmatrix} = \frac{1}{\ell} \sum_{\mathbf{x} \in s} \begin{bmatrix} \phi_\mathrm{M}(\mathbf{x}) \\ \phi_\mathrm{S}(\mathbf{x}) \end{bmatrix} \tag{4.14}$$

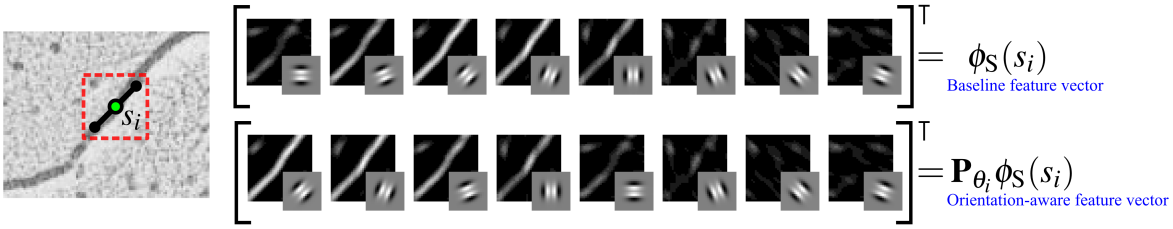


Figure 4.8: The baseline steerable feature vector $\phi_S(s_i)$ is created by steerable filtering responses $(\mathbf{F}_\theta * I)$ as the increasing order of orientation. To take into account the direction of line segment s_i, we multiply permutation matrix P_{θ_i} by the baseline $\phi_S(s_i)$.

4.3.2 Permutation of feature maps

As the result of ranking learning system, we obtain a model parameter $\mathbf{w}$ that determines the relative importance of each element in the feature vector. If the training set consists of a particular orientation, the corresponding parameter has a larger value than others. However, such situation can induce an over-fitting. To ease this issue, we also exploit the own directional information of the line segments. For this purpose, we permute the order of elements in the baseline steerable feature vector $\phi_S(s)$ according to the orientation of the given line segment s.

Let $\mathbf{P}_{\theta_i}$ be a permutation matrix that shuffles the order of the feature vector according to the given orientation θ_i:

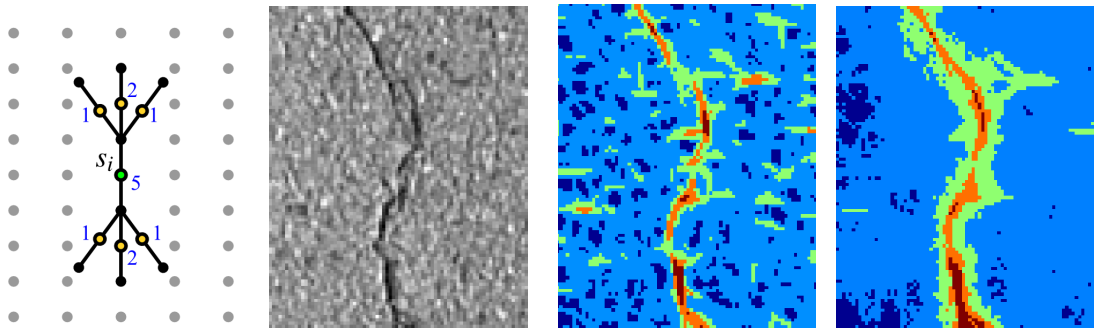
$$\mathbf{P}_{\theta_i} = \left[\mathbf{e}_i \ \mathbf{e}_{(i+n)\circ k} \ \mathbf{e}_{(i-n)\circ k} \ \cdots \ \mathbf{e}_{(i+\lfloor k/2 \rfloor)\circ k}\right]^\mathsf{T}, \quad n = 1, 2, \cdots, \lfloor k/2 \rfloor - 1,$$

where $\mathbf{e}_i$ denotes a row vector with 1 in the i-th element and 0 in the others. We define a circular modulation operator $\circ$ which is given by:

$$i \circ k = ((i-1) \bmod k) + 1. \tag{4.15}$$

For instance, $\mathbf{P}_{\theta_4}$ for $k = 8$ is defined as

$$\mathbf{P}_{\theta_4} = \begin{bmatrix} \mathbf{e}_4 \\ \mathbf{e}_5 \\ \mathbf{e}_3 \\ \mathbf{e}_6 \\ \mathbf{e}_2 \\ \mathbf{e}_7 \\ \mathbf{e}_1 \\ \mathbf{e}_8 \end{bmatrix} = \begin{bmatrix} 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}.$$



(a) (b) (c) (d)

Figure 4.9: The topology of the spatial feature grouping is depicted in (a), where the corresponding weighting factors $\omega_j, j \in \mathcal{N}_i$ of the neighboring line segments are denoted in blue colored numbers. The spatial grouping of the curvilinear features is able to improve the local consistency of the output rankings. Given image (b) is corrupted by irregular background textures. We visualize the output rankings: (c) is obtained without spatial feature grouping, whereas (d) shows the output rankings with the proposed spatial feature grouping.

To compute the orientation-aware feature vectors $\phi'(s)$, we apply the permutation matrix $\mathbf{P}_\theta$ to re-order the elements of the baseline feature vector $\phi(s)$. For a better understanding, we give a toy example in Figure 4.8 how to shuffle the baseline feature vector for the given orientation. Suppose that the orientation of line segments s_2 and s_5 are associated with θ_2 and θ_5, respectively. Then, the orientation-aware feature vectors $\phi'(s_2)$ and $\phi'(s_5)$ are given by

$$\phi'(s_2) = \begin{bmatrix} \phi_{\mathrm{M}}(s_2) \\ \mathbf{P}_{\theta_2}\phi_S(s_2) \end{bmatrix}, \quad \phi'(s_5) = \begin{bmatrix} \phi_{\mathrm{M}}(s_5) \\ \mathbf{P}_{\theta_5}\phi_S(s_5) \end{bmatrix}.$$

Now the elements of the orientation-aware feature vectors are aligned in the order of dominant directional response for the given line segments. Therefore, the model parameter $\mathbf{w}$ of the learning system properly works to measure compatibility of input line segments and the given image data. Figure 4.8 compares the baseline feature vector with the orientation-aware feature vector for the given line segment s_i.

4.3.3 Spatial grouping of the features

To improve segmentation performance, the piecewise smoothness assumption on the image grid is often used [Geman 84, Boykov 04], *e.g.*, 4-connected neighboring pixels have the same label with a high probability. While the thin line structure shows

the internal similarity, such prior model also encourages background textures to be classified as the same segment so that it produces discontinuities on the curvilinear structure. Instead, we propose a new topology to consider the neighborhood system of line segments. The proposed neighborhood system is designed to realize a smoothly connected curvilinear structure. Thus, to enhance the spatial coherence of the output rankings, the neighboring set consists of shifted line segments to the extendable direction and shifted-and-rotated line segments with a small curvature. (see Figure 4.9 (a)). We define the spatial grouping operator $\bar{\phi}$ as:

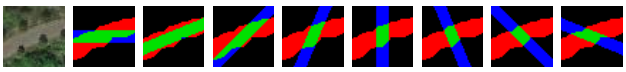
$$\bar{\phi}(s_i) = \frac{\sum_{j \in \mathcal{N}_i} \omega_j \phi'(s_j)}{\sum_{j \in \mathcal{N}_i} \omega_j}, \tag{4.16}$$

where $\mathcal{N}_i$ denotes the neighboring set of the line segment s_i and ω_j is the corresponding weight factor (Figure 4.9 (a) details the values of ω_j used in this work).

Figure 4.9 compares the output rankings obtained by individual feature vector and spatially grouped feature vector. When we learn a ranking function without the spatial feature grouping operation, the output rankings are less reliable due to the corrupted background texture. On the other hand, the output rankings with the spatial feature grouping increase the rankings of line segments corresponding to the latent curvilinear structures.

4.4 Learning

In this section, we develop a ranking learning system based on SSVM [Tsochantaridis 04]. Our goal is to learn a function $h : \mathcal{S} \mapsto \mathcal{Z}$ that predicts structured output rankings of the line segments. For the setup of machine learning system, a training dataset $\mathcal{D} = \{(s_i, z_i)\}_{i=1}^{K}$ is required which consists of the input-and-output pairs. Assume that we have a list of line segments $\mathbf{s} = \{s_1, \ldots, s_K\} \subseteq \mathcal{S}$ and the relevant ranking values $\mathbf{z} = \{z_1, \ldots, z_K\} \subseteq \mathcal{Z},\ z_i \in [0,\ 1]$. The line segments $\mathbf{s}$ for training are independently sampled from the images of the training dataset. To determine the corresponding rankings $\mathbf{z}$ of the input line segments is a tricky part in this work. Recall that the function g represents a binary map to classify pixels on the latent curvilinear structure. We use the binary map as the ground truth of the curvilinear structure. However, the ground truth g cannot encode the shape information of the line segments (length, orientation, and thickness). In other words, line segments proposed at pixel $\mathbf{x}$ could be irrelevant with the true curvilinear structure due to the shape information even though the pixel is on the curvilinear structure,*i.e.*, $g(\mathbf{x}) = 0$. We evaluate the shape dissimilarity between the given line



	0°	22.5°	45°	67.5°	90°	112.5°	135°	157.5°
F_1	0.56	**0.78**	0.51	0.30	0.27	0.22	0.19	0.26
z	0.44	**0.22**	0.49	0.70	0.73	0.78	0.81	0.74

Figure 4.10: Example of the evaluation of the given line segment s_i and corresponding ground truth patch g_i. Blue and red pixels show superfluous and missed areas, respectively, whereas green pixels paint the agreed area between line segment and ground truth.

segments s_i and the corresponding patch of the ground truth map g_i:

$$z_i = 1 - \texttt{score}(s_i, g_i). \tag{4.17}$$

Many options, such as correlation coefficient [Jeong 15a], Jaccard index [Jaccard 12], and Hamming distance [Hamming 50], are available to compute the function `score`. In this work, we employ F_1 measure [Martin 04, Arbelaez 10] which is defined as

$$\texttt{score}(s_i, g_i) = F_1(s_i, g_i) = \frac{2P_iR_i}{P_i + R_i}, \tag{4.18}$$

where P_i and R_i denote *precision* and *recall*, respectively [Philbin 07, Manning 08], which are associated with the line segment s_i. Figure 4.10 shows an example that computes the shape similarity score of a line segment and the corresponding ground truth patch. If s_i lies on the true curvilinear structure g_i, the corresponding ranking value z_i is closer to 0; otherwise s_i is on a non-curvilinear structure, z_i is set to 1.

We use a joint feature map of input-and-output pairs $\Psi(\mathbf{s}, \mathbf{z}) : \mathcal{S} \times \mathcal{Z} \mapsto \mathbb{R}^N$ to obtain structured output rankings. A scoring function $h : \mathcal{S} \times \mathcal{Z} \mapsto \mathbb{R}$ is defined as the inner product of the joint feature vector and a model parameter $\mathbf{w}$. For a given line segment, a prediction $\hat{\mathbf{z}} = h(\mathbf{s}; \mathbf{w})$ is performed by finding rankings that maximize the scoring function:

$$\hat{\mathbf{z}} = \underset{\mathbf{z} \in \mathcal{Z}}{\operatorname{argmax}}\, h(\mathbf{s}, \mathbf{z}; \mathbf{w}) = \mathbf{w}^\intercal \Psi(\mathbf{s}, \mathbf{z}). \tag{4.19}$$

We define the joint feature map $\Psi(\mathbf{s}, \mathbf{z})$ to obtain the decreasing order of rankings among input line segments:

$$\Psi(\mathbf{s}, \mathbf{z}) = \sum_{ij} z_{ij} \left(\bar{\phi}(s_i) - \bar{\phi}(s_j) \right), \tag{4.20}$$

where the ranking matrix z_{ij} indicates the pairwise order of the objects s_i and s_j in terms of shape dissimilarity measure (4.17). In other words,

$$z_{ij} = \begin{cases} +1 & \text{if } z_i > z_j, \\ -1 & \text{otherwise.} \end{cases}$$

The parameter of the learning system $\mathbf{w} \in \mathbb{R}^N$ is obtained by minimizing the following constrained objective function:

$$\begin{array}{ll} \min\limits_{\mathbf{w},\xi} & \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{K}\sum_{i=1}^{K}\xi_i \\ \text{subject to} & \mathbf{w}^\intercal\Psi(\mathbf{s},\mathbf{z}) \geq \mathbf{w}^\intercal\Psi(\mathbf{s},\hat{\mathbf{z}}) + \Delta(\mathbf{z},\hat{\mathbf{z}}) - \xi; \quad \forall\hat{\mathbf{z}},\hat{\mathbf{z}} \neq \mathbf{z} \end{array} \tag{4.21}$$

where the first term of (4.21) penalizes the complexity of the solution (regularization) and the second term measures the relaxed loss of the system to fit the parameter into training data (regression). C controls the relative importance between the two terms. $\Delta(\mathbf{z},\hat{\mathbf{z}})$ is a loss function that counts the number of incorrectly ranked pairs. To solve this optimization problem, we use the cutting plane algorithm ([Joachims 09], see Algorithm 4 in Section 4.1.2).

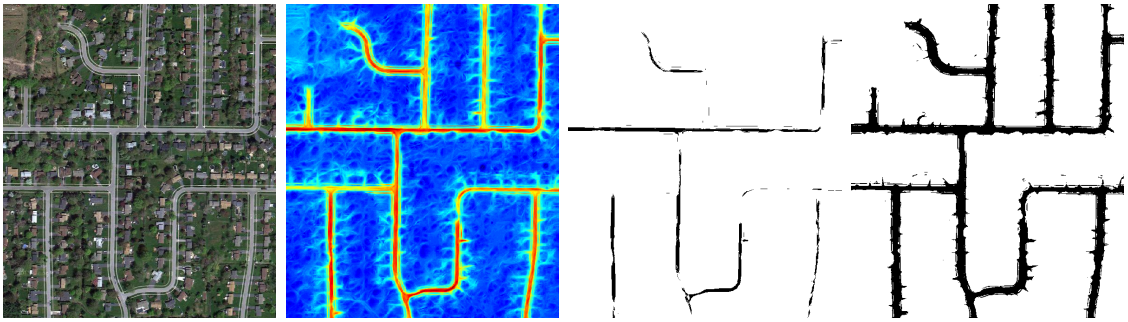
We have obtained a ranking function $h(\mathbf{s};\mathbf{w})$ that arranges the line segments in the order of correspondence with the underlying curvilinear structure. Specifically, the linear combination of the model parameter $\mathbf{w}$ and the proposed feature descriptor $\bar{\phi}(s)$, $\hat{z} = \mathbf{w}^\intercal\bar{\phi}(s)$, is used to evaluate the line segment s. At each pixel grid, we create k line segment candidates, where k denotes the number of orientations. We then estimate the local orientation using the ranking function:

$$\hat{i} = \underset{i\in\{1,\ldots,k\}}{\operatorname{argmax}}\, \mathbf{w}^\intercal\bar{\phi}(s(\theta \leftarrow \theta_i)), \text{ then } \theta_{\max} = \theta_{\hat{i}}, \tag{4.22}$$

where $s(\theta \leftarrow \theta_i)$ denotes a candidate of the line segment which updates its orientation for θ_i whereas the rest of the parameters (center locations, length, and thickness) are fixed. Moreover, the ranking function is employed to indicate a pixel on the curvilinear structures. For this purpose, the output rankings are computed at each pixel grid using the line segment maximizing local orientation responses via (4.22).

4.5 Inference

In this section, we aim to develop a graphical model to infer the topology of the curvilinear structure. The main motivation to represent the curvilinear structure as a graph is to use an optimal number of pixels to reconstruct the curvilinear structure without redundancies. Also, we want to see the topological features of the structure



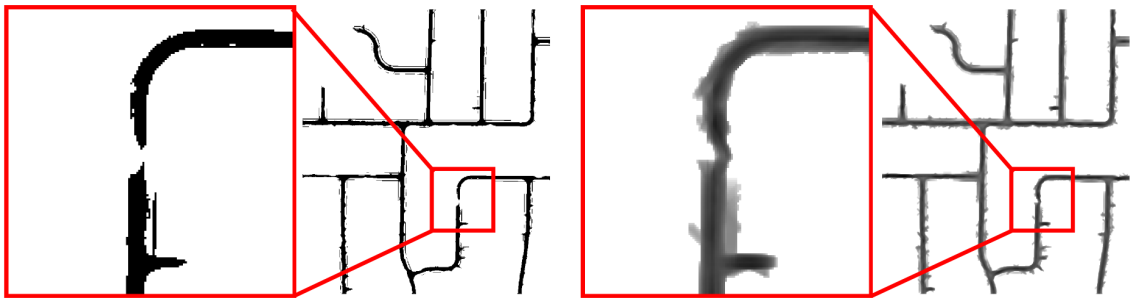
(a) Input (b) Output rankings (c) 4.16% (d) 14.78%

Figure 4.11: Binary segmentation maps are built depending on statistics of the output rankings.

at different levels of detail.

For the sake of convenience, we recall several terminologies related to graph theory before developing the proposed algorithm. We consider the graph $G = (V, E)$ where V is the set of pixels. Each pixel being given its image coordinates, and pixels $\mathbf{u} = (i, j)$ and $\mathbf{v} = (m, n)$ are adjacent if and only if $|i - m| \leq 1$ and $|j - n| \leq 1$. Moreover, we assign a weight for each edge $\{\mathbf{u}, \mathbf{v}\} \in E$ as $w(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \frac{\hat{z}_\mathbf{u} + \hat{z}_\mathbf{v}}{2}$, where $\|\mathbf{u} - \mathbf{v}\|$ denotes the Euclidean distance between pixel coordinates of $\mathbf{u}$ and $\mathbf{v}$. $\hat{y}_\mathbf{u}$ and $\hat{y}_\mathbf{v}$ are the output rankings at $\mathbf{u}$ and $\mathbf{v}$, respectively. A path P in the graph G is a sequence of distinct vertices such that consecutive vertices are adjacent. The length of P is the sum of the weight of its edges, and the distance $dist(\mathbf{u}, \mathbf{v})$ between two vertices $\mathbf{u}$ and $\mathbf{v}$ is the minimum length of a path from $\mathbf{u}$ to $\mathbf{v}$. The eccentricity $ecc(\mathbf{v})$ denotes the maximum distance from the node $\mathbf{v}$ to a node $\mathbf{u} \in V$, *i.e.,* $ecc(\mathbf{v}) = \max_{\mathbf{u} \in V} dist(\mathbf{u}, \mathbf{v})$. The diameter $diam(G)$ of G equals $\max_{\mathbf{v} \in V} ecc(\mathbf{v})$, *i.e.,* it is the maximum distance between two nodes in G.

Intuitively, to infer the topology of the curvilinear structure, we look for long shortest paths in the subgraph G' of G induced by the nodes with high rankings. More precisely, our algorithm computes a diameter of G', *i.e.,* a shortest path P with length $diam(G')$. This path P is added in the curvilinear structure inference, then the path P is contracted into a single node (equivalently, the weight of all edges of P become 0), and the process is repeated till the diameter of the subgraph is smaller than pre-defined path length $\ell_{\min}$. The entire procedure is summarized in Algorithm 5.



(a) Binary segmentation map (b) Dissimilarity score map

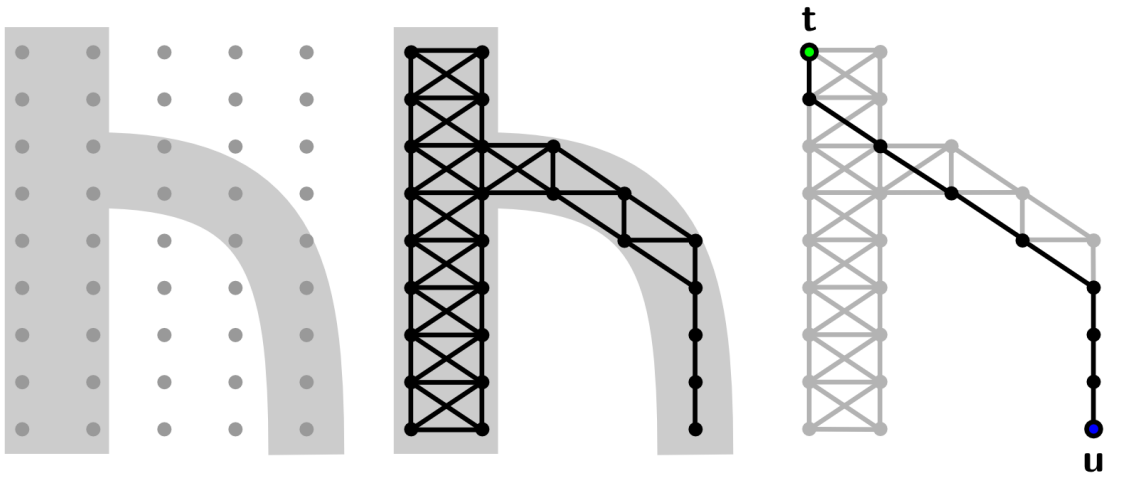
Figure 4.12: A binary segmentation map is created by thresholding output rankings at a deterministic value, where the value is obtained by the statistics of pixels on the curvilinear structure from the training dataset. While the structured output rankings contain the shape information of the line segments, we cannot exploit such useful information based on the binary segmentation map. The dissimilarity score map is generated by re-aggregation of the binary segmentation map. For each pixel on the binary segmentation map, we introduce the shape information which is already embedded in the output rankings. Then, we compute dissimilarity scores of the disassembled line segments and the binary segmentation map. Finally, we fill areas corresponding to the line segments with the dissimilarity scores.

Algorithm 5 Iterative curvilinear path reconstruction

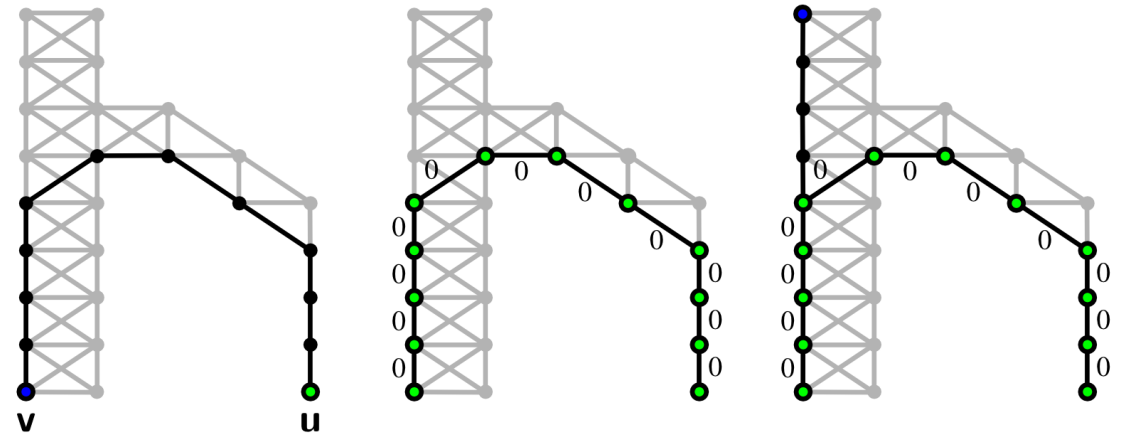
Inputs:
$G' = (V', E') \sim$ an undirected and weighted subgraph of the image G; and
$\ell_{\text{min}} \sim$ a minimum length of the curvilinear structure
Output:
$P \sim$ a set of vertices corresponding to the curvilinear structure
$P \leftarrow \emptyset$
`longest_path_length` $\leftarrow |V'|$
while do
  Compute the longest path Q in G' using 2-sweep algorithm
  `longest_path_length` $\leftarrow |Q|$
  if `longest_path_length`$< \ell_{\text{min}}$ **then**
    break
  end if
  $P \leftarrow P \cup Q$
  $w(\mathbf{u}, \mathbf{v}) \leftarrow 0, \forall \mathbf{u}, \mathbf{v} \in Q$, All edge weights on the path Q become 0
end while

The first part of the algorithm is to extract the subgraph G' from the whole set of pixels. This subgraph should contain the curvilinear structure and be relatively small for an efficient computation. For this purpose, from the training datasets, we first compute the average proportion ρ of pixels being part of the curvilinear structure. Then, the binary segmentation map is induced by $\rho|I|$ pixels according to the output rankings, related to a high ranking. However, due to the noises and the choice of a small ρ, the binary segmentation map may be disconnected so that some parts do not reflect the connected components of the curvilinear structure. Moreover, the binary segmentation map only encodes the center coordinates of the line segments (see Figure 4.11) while the output rankings contain the shape information of the line segments. To overcome these issues, we disjoint the binary segmentation map into line segments, and compute the dissimilarity scores (4.17) between these line segments and the corresponding patches of the binary segmentation map. We fill up areas corresponding to the line segments with those of dissimilarity scores. Specifically, for overlapping areas between adjacent line segments, we assign the average value of theirs (see Figure 4.12). Thus, the subgraph G' is built upon a subset of pixels $V' \subset V$ which corresponds to the dissimilarity score map. We also update the weights for each edge $\{\mathbf{u}, \mathbf{v}\} \in E$ in the subgraph as $w(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \frac{b_{\mathbf{u}}\hat{z}_{\mathbf{u}} + b_{\mathbf{v}}\hat{z}_{\mathbf{v}}}{2}$, where $b_{\mathbf{u}}$ and $b_{\mathbf{v}}$ denote the dissimilarity scores at $\mathbf{u}$ and $\mathbf{v}$, respectively.

The time-consuming part of the proposed algorithm is the computation of a diameter of G'. Rather than computing all pair distances (which requires a linear number of application of Dijkstra's algorithm), we use an efficient heuristic algorithm called *2-sweep algorithm* proposed by [Corneil 01]. The 2-sweep algorithm randomly picks a vertex $\mathbf{t}$ in G', then performs Dijkstra's algorithm from $\mathbf{t}$ to find a node $\mathbf{u}$ at the maximum distance from $\mathbf{t}$, *i.e.*, $dist(\mathbf{u}, \mathbf{t}) = ecc(\mathbf{t})$. Then, it computes (using Dijkstra's algorithm) a path from $\mathbf{u}$ to a node $\mathbf{v}$ at the maximum distance from $\mathbf{u}$. The length of the second path (from $\mathbf{u}$ to $\mathbf{v}$) is a good estimation of $diam(G')$. Note that this algorithm is able to compute the exact diameter if the graph has tree structure [Borassi 15]. Topologically speaking, most of latent curvilinear structures are very close to trees, so that the use of the 2-sweep algorithm is well adapted. For a better understanding, we schematically explain the intermediate steps of the proposed curvilinear structure inference algorithm in Figure 4.13.



(a) (b) (c)



(d) (e) (f)

Figure 4.13: Toy example of the proposed curvilinear structure inference algorithm: (a) input image contains a curvilinear structure which is denoted by gray color; (b) subgraph G' is induced from the structured output rankings; (c) and (d) show the intermediate processes of the 2-sweep algorithm starting from node **t** to find a diameter of the subgraph; (e) we assign 0 weight for all edges on the path; and (f) we repeat the process and add branches if the path length is larger than pre-defined length $\ell_{\min}$.

4.6 Experimental results

In this section, we first discuss the parameters of the proposed algorithm according to the datasets. We then compare the quantitative and qualitative results of the proposed algorithm and those of competing models proposed by [Frangi 98], [Law 08], [Becker 13], [Sironi 14], and [Jeong 15a].

4.6.1 Parameters

The proposed algorithm requires few parameters to compute the curvilinear feature descriptor ϕ for defining physics of the line segment,*i.e.,* length ℓ and thickness τ, and to train a ranking function via SSVM. For the morphological profiles ϕ_M, we apply the path opening operator 4 times with different length parameters L. We linearly increase such length parameters by adding η from the minimum length L^*. Thus, $L \in \{L^* + n\eta | n = 0, 1, 2, 3\}$. In the case that the curvilinear structure appears darker than surrounding pixels, we invert the input images before applying the morphological operation. This is because the path opening operator assumes pixels on the curvilinear path to be brighter than neighboring pixels. For the steerable features ϕ_S, we perform convolution of the steerable filters with the normalized input images. The steerable filter banks consist of 3 basis kernels with $k = 8$ different orientations, where the size of the filters is 21×21 (see Figure 4.7). To build the basis kernels, we superpose the elongated Gaussian kernels using different aperture and smoothing factors. The aperture value determines the space between the Gaussian kernels with respect to the minor axis. We control the aperture size of the Gaussian kernels as $a = \{6, 3, 1\}$ to take into account the thickness of the curvilinear structure. The smoothing factor of the outer kernels σ_o^2 is set to 2.5 and those of inner kernels σ_i^2 is set to 1.5. The elongation factor γ is fixed as 2.5 for all experiments. Note that the steerable filters are symmetric with respect to the major axis. Also, we assume that the length of line segment should be larger than its thickness, $\ell > \tau$, and these two values depend on the dataset. For SSVM training, we sample 500 positive and negative query line segments, respectively, for every training image. C which controls the relative importance of regularization and regression terms in (4.21) is set to 0.01 for all datasets. We choose the set of parameters for the SSVM training via 3-fold cross validation [Kohavi 95] which maximizes the average F_1 score of the training set.

We test our curvilinear structure model on the following public datasets:

- **DRIVE** [Staal 04]: This dataset consists of 40 retina scan images with manual segmentation by ophthalmologists to evaluate the blood vessel segmentation algorithms. We use 20 images for the training and 20 images for the test, re-

spectively. We build the morphological profiles with $(L^*, \eta) = (40, 10)$ and use $(\ell, \tau) = (5, 3)$.

- **RecA** [Jeong 15b]: We collect electron microscopic images of RecA proteins on DNA which contain filament structure. We use 4 training images and 2 test images. We set $(L^*, \eta) = (40, 10)$, and $(\ell, \tau) = (5, 3)$, respectively.

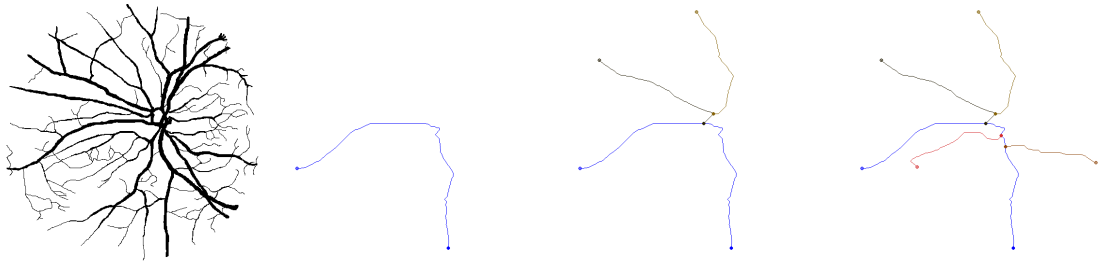
- **Aerial** [Sironi 14]: The dataset contains 14 remote sensing images of road networks. We select 7 images for the training phase and 7 images for the testing phase, respectively. (L^*, η) is set to $(50, 50)$ and each line segment consists of $\ell = 15$ and $\tau = 4$.

- **Cracks** [Chambon 10]: Images of the dataset correspond to road cracks on asphalt surfaces. We use 6 images to train and test the algorithms on different 6 images. Parameters for the morphological profiles (L^*, η) are set to $(10, 5)$, and for the physics of line segments (ℓ, τ) are set to $(5, 3)$.

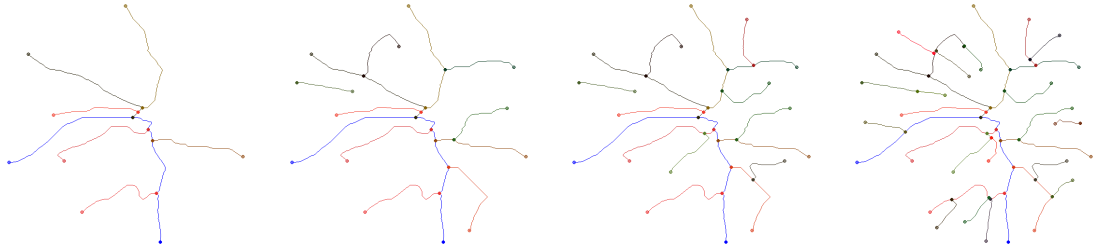
To create subgraph G' for the curvilinear structure inference, we compute the average proportion ρ of pixels on the curvilinear structure by observing the training images. We obtain ρ values for DRIVE, RecA, Aerial and Cracks dataset as 8.63%, 4.99%, 8.16%, and 4.3%, respectively. The minimum path length $\hat{\ell}$ is adaptively set to 40, 30, 80, and 30 for DRIVE, RecA, Aerial and Cracks datasets, respectively, to obtain the best performance.

4.6.2 Evaluations

If we use pre-computed feature descriptors, sorting the given line segments takes less than a seconds. The proposed algorithm progressively reconstructs the curvilinear structure by adding a long path on the subgraph. Figure 4.14 and Figure 4.15 show the intermediate steps of the proposed graph inference algorithm for Drive and Aerial dataset, respectively. Unlike the previous models, the proposed algorithm is able to show different levels of detail for the latent curvilinear structure. Such information, which visualizes shape complexity of the curvilinear structure, cannot be retrieved by tuning the threshold. In practice, a few number of iterations is required to converge the proposed algorithm, and each step to find a long path takes less than milliseconds for the computation. For the experiments, we use a PC with a 2.9 GHz CPU (4 cores) and 8 GB RAM. Computing curvilinear feature descriptors is the most time consuming part. To compute all possible line segments in 800×500 pixels image, it takes 5 minutes – we expect that parallel implementation can dramatically reduce the computation time. To train SSVM, we use 1000 examples and it takes a minute.



(a) Ground truth (b) # 1 (c) # 3 (d) # 5

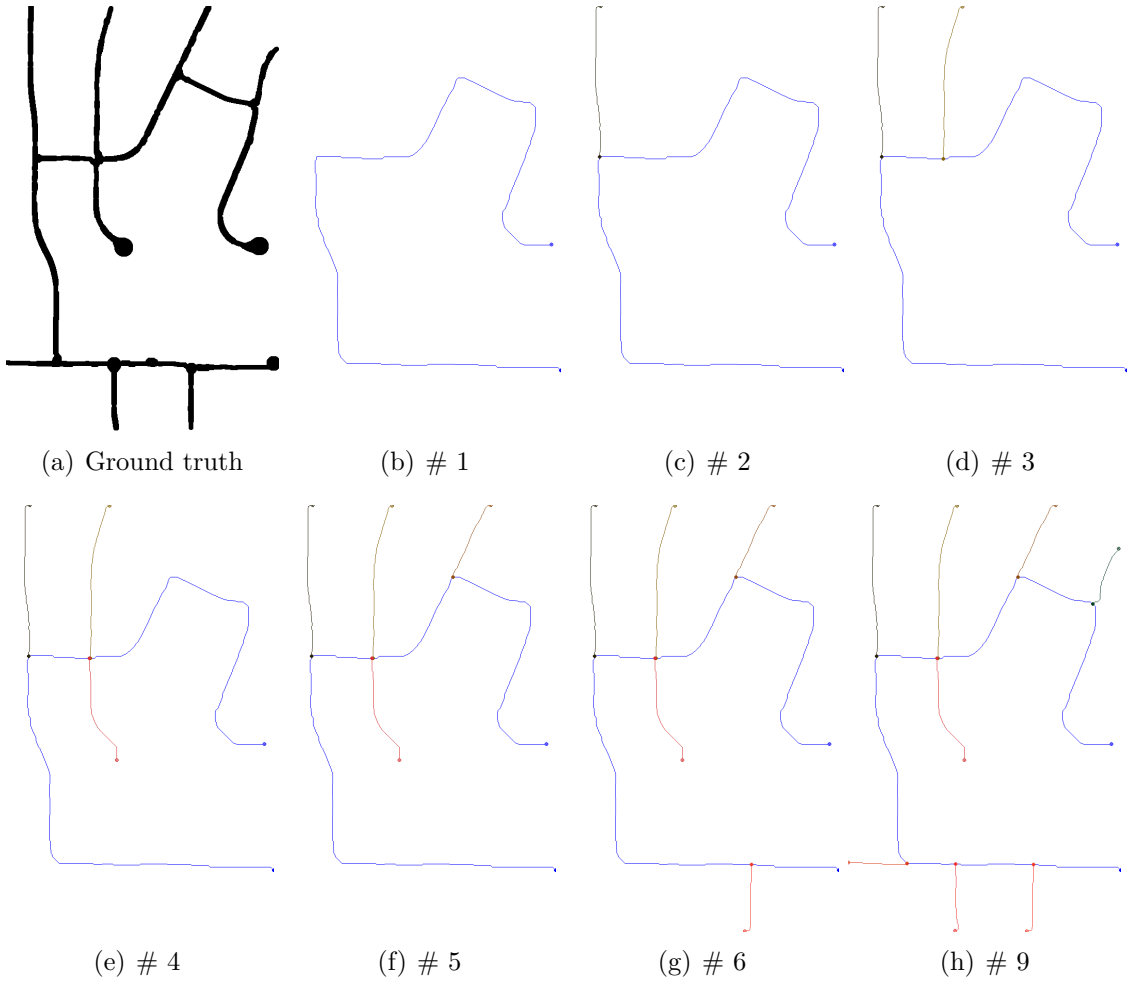


(e) # 7 (f) # 12 (g) # 16 (h) # 28

Figure 4.14: Intermediate steps of the curvilinear structure reconstruction for a retina image. We iteratively reconstruct the curvilinear structure according to topological importance orders. As the iteration goes on, detail structures (layers) appear.

We visually compare the performance of the proposed algorithm with the competing algorithms. Figure 4.16, 4.17, 4.18, and 4.19 show the results from DRIVE, RecA, Aerial, and Cracks dataset, respectively. Although the proposed algorithm is the most suitable to show the topological importance level of the curvilinear structures, its intermediate result (dissimilarity score map) is comparable to the results of curvilinear segmentation algorithms.

For the quantitative evaluation, we provide precision-and-recall curve of the proposed algorithm (dissimilarity score map) and the state-of-the-art models in Figure 4.20. Moreover, Table 4.1 summarizes the Mean Average Precision (MAP) scores [Philbin 07, Jégou 08, Arandjelović 11] corresponding to Figure 4.20 for each dataset. We plot the curves by adjustment of the threshold value to represent the curvilinear structure. The dissimilarity score map, which is an intermediate result of the proposed algorithm, is used to plot Figure 4.20 and to compute Table 4.1. The measure of true positive is sensitive for the misalignment; therefore, we consider surrounding pixels of the detection results as the true positive if a predicted point is falling into the ground truth with a small radius. To provide the average performance of each dataset, we compute the interpolated average precision values at the fixed recall values,



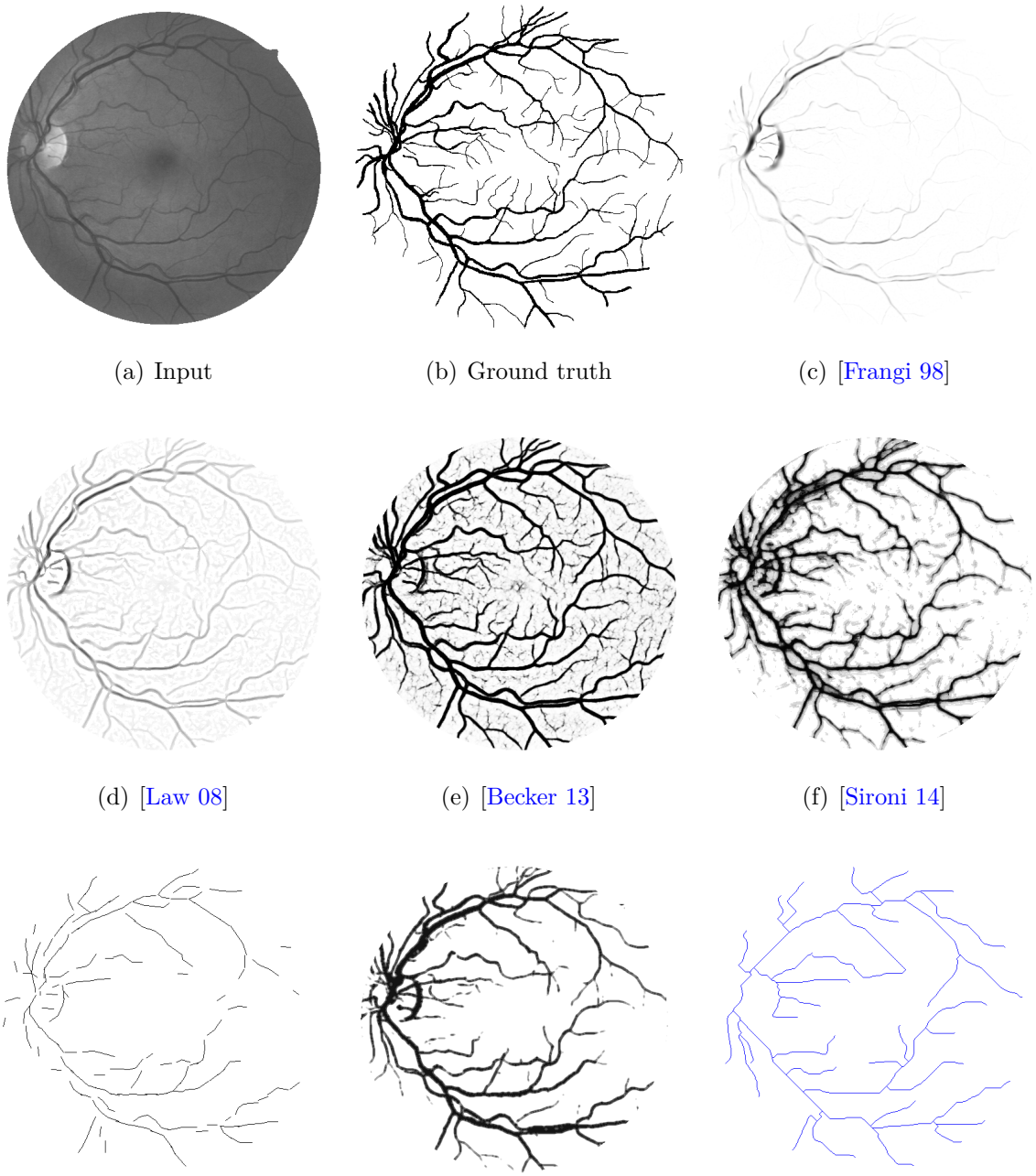
(a) Ground truth (b) # 1 (c) # 2 (d) # 3

(e) # 4 (f) # 5 (g) # 6 (h) # 9

Figure 4.15: Intermediate steps of the curvilinear structure reconstruction for a road network image.

where the method is detailed in [Manning 08]. However, the evaluation with precision and recall is unable to measure the structural features. We also provide the average proportion of pixels to represent the curvilinear structures and the corresponding F_1 scores in Figure 4.21 and Table 4.2. The graph shows that the proposed algorithm efficiently draws the curvilinear structures using a smaller number of pixels than the other algorithms. In Table 4.2, we observe that the proposed algorithm outperforms than competing algorithms for all datasets, except Drive dataset. It is because the proposed algorithm uses the fixed thickness τ to describe line segments whereas the images of Drive dataset consists of blood vessels with varied thicknesses. The performance could be improved if we consider the thickness of the line segment as a learnable variable.

It is worth to mention the over-fitting problem in curvilinear segmentation task

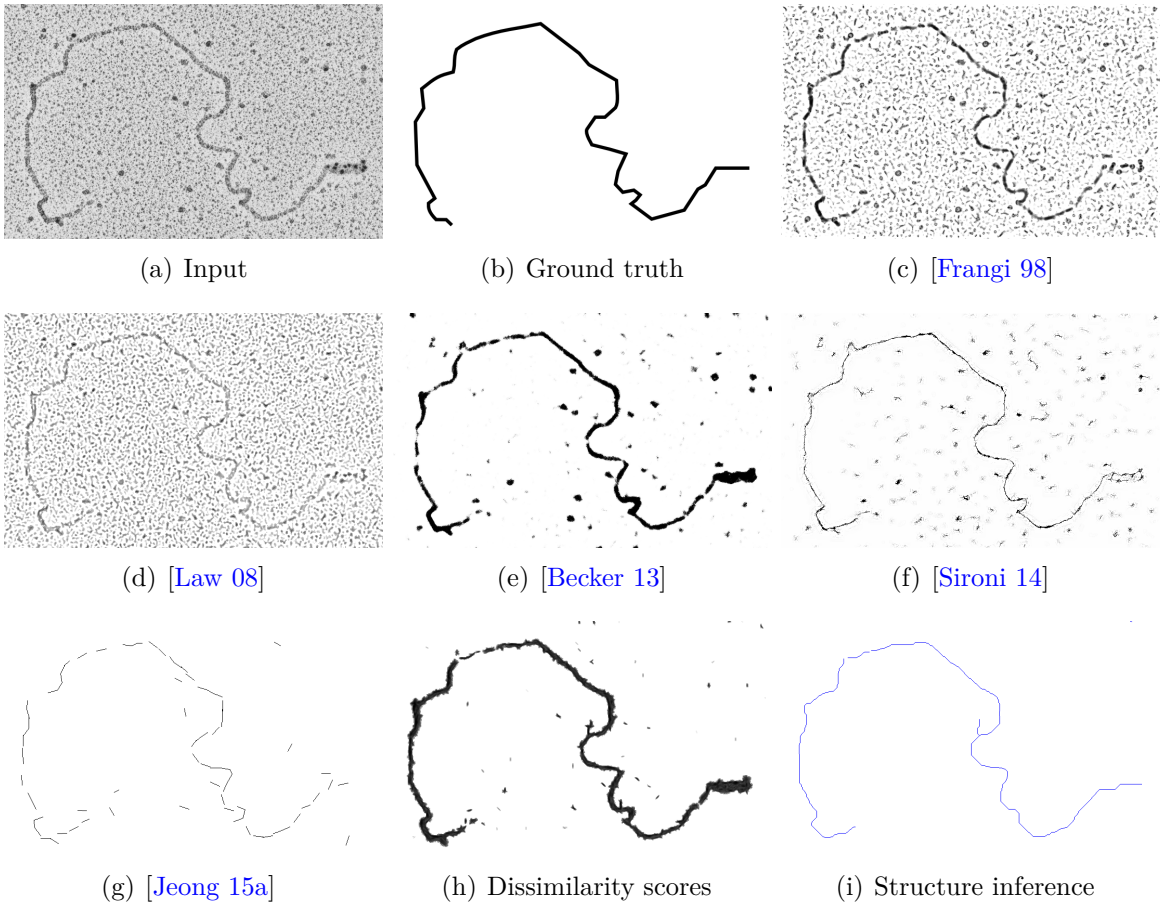


(a) Input (b) Ground truth (c) [Frangi 98]

(d) [Law 08] (e) [Becker 13] (f) [Sironi 14]

(g) [Jeong 15a] (h) Dissimilarity scores (i) Structure inference

Figure 4.16: Experimental results on **DRIVE** dataset



(a) Input (b) Ground truth (c) [Frangi 98]

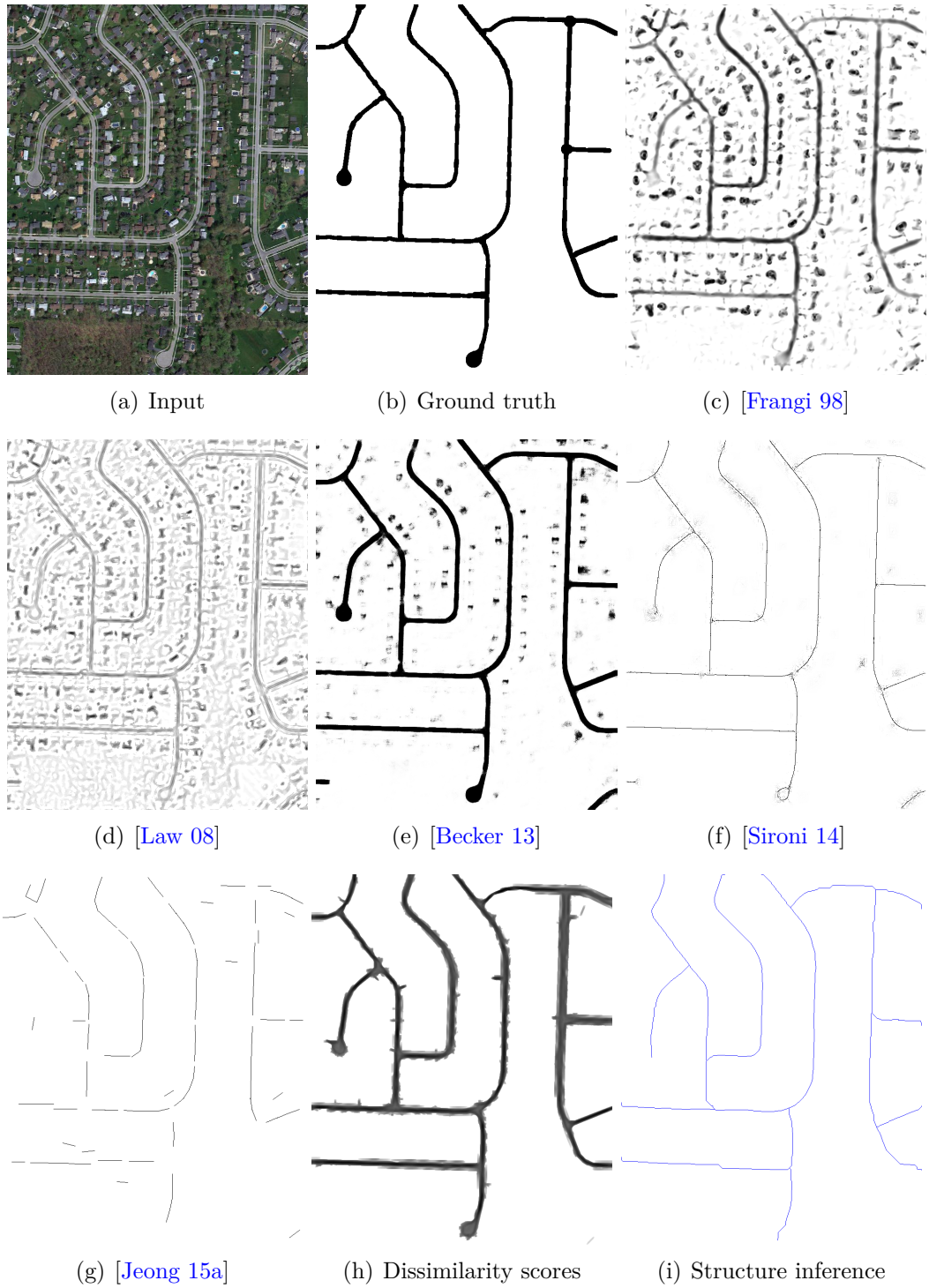
(d) [Law 08] (e) [Becker 13] (f) [Sironi 14]

(g) [Jeong 15a] (h) Dissimilarity scores (i) Structure inference

Figure 4.17: Experimental results on **RecA** dataset

Table 4.1: Mean Average Precision (MAP) scores of each dataset for quantitative evaluation. Boldfaced numbers are used to indicate the best performance in each test.

	DRIVE	RecA	Aerial	Cracks
[Frangi 98]	0.61	0.55	0.38	0.074
[Law 08]	0.73	0.33	0.25	0.14
[Becker 13]	**0.83**	0.70	0.88	**0.37**
[Sironi 14]	0.68	0.64	**0.89**	0.36
Proposed	0.67	**0.77**	0.76	0.25

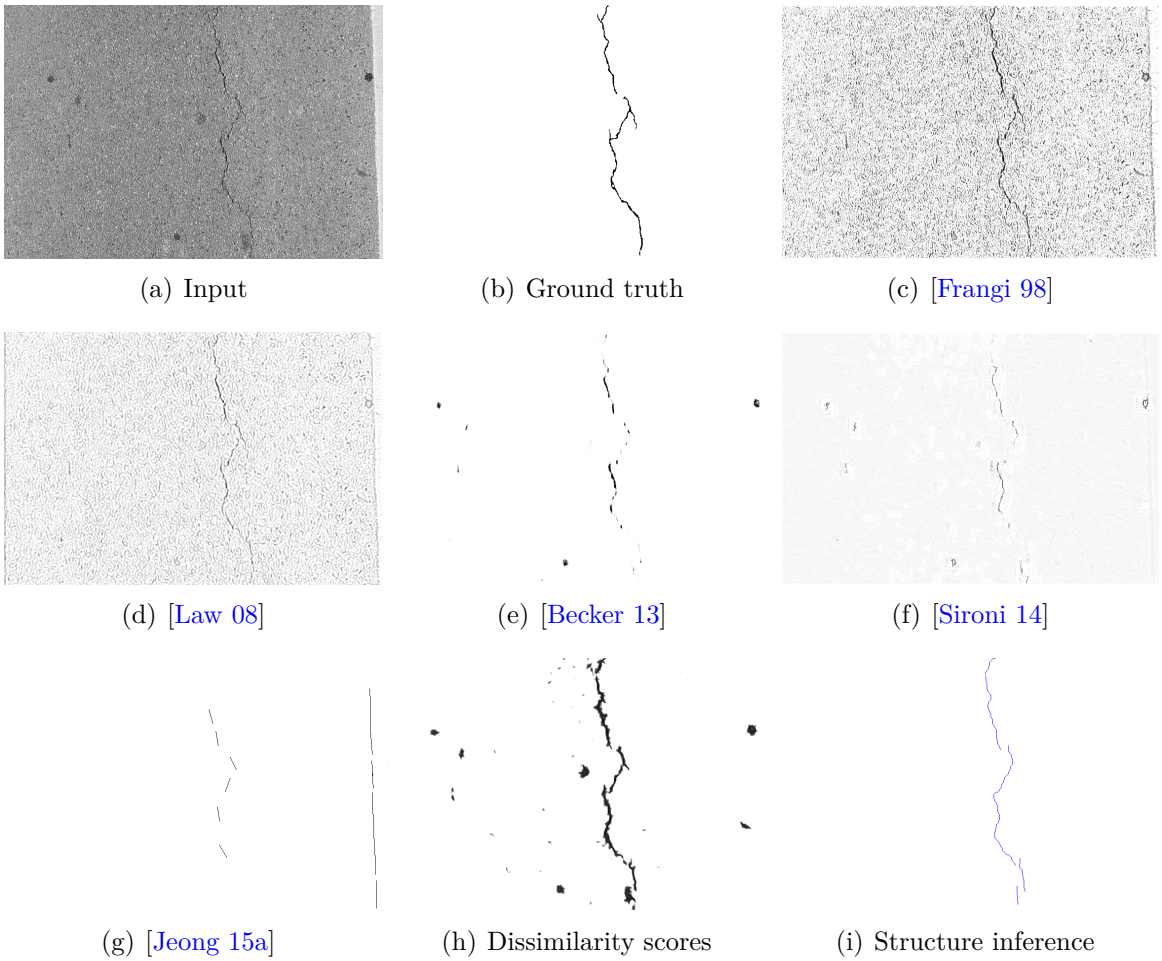


(a) Input (b) Ground truth (c) [Frangi 98]

(d) [Law 08] (e) [Becker 13] (f) [Sironi 14]

(g) [Jeong 15a] (h) Dissimilarity scores (i) Structure inference

Figure 4.18: Experimental results on **Aerial** dataset



(a) Input (b) Ground truth (c) [Frangi 98]

(d) [Law 08] (e) [Becker 13] (f) [Sironi 14]

(g) [Jeong 15a] (h) Dissimilarity scores (i) Structure inference

Figure 4.19: Experimental results on **Crack** dataset

Table 4.2: F_1 scores and the average proportion of the pixels being a part of curvilinear structure (%) are computed. Boldfaced numbers are used to show the best F_1 score in each test. Gray colored cells in this table denote the algorithm using the minimum number of pixels to represent the curvilinear structure.

	DRIVE	RecA	Aerial	Cracks
[Frangi 98]	0.33 / 37.09%	0.33 / 16.14%	0.32 / 22.19%	0.056 / 28.64%
[Law 08]	0.43 / 19.60%	0.21 / 29.82%	0.25 / 29.51%	0.085 / 33.00%
[Becker 13]	0.50 / 12.67%	0.45 / 9.39%	0.53 / 9.76%	0.23 / 32.13%
[Sironi 14]	0.55 / 5.41%	0.50 / 5.47%	0.55 / 0.83%	0.27 / 17.12%
[Jeong 15a]	**0.59** / 0.97%	0.57 / 0.35%	0.47 / 0.46%	0.12 / 0.34%
Proposed	0.36 / 1.07%	**0.59** / 0.35%	**0.59** / 0.57%	**0.38** / 0.17%

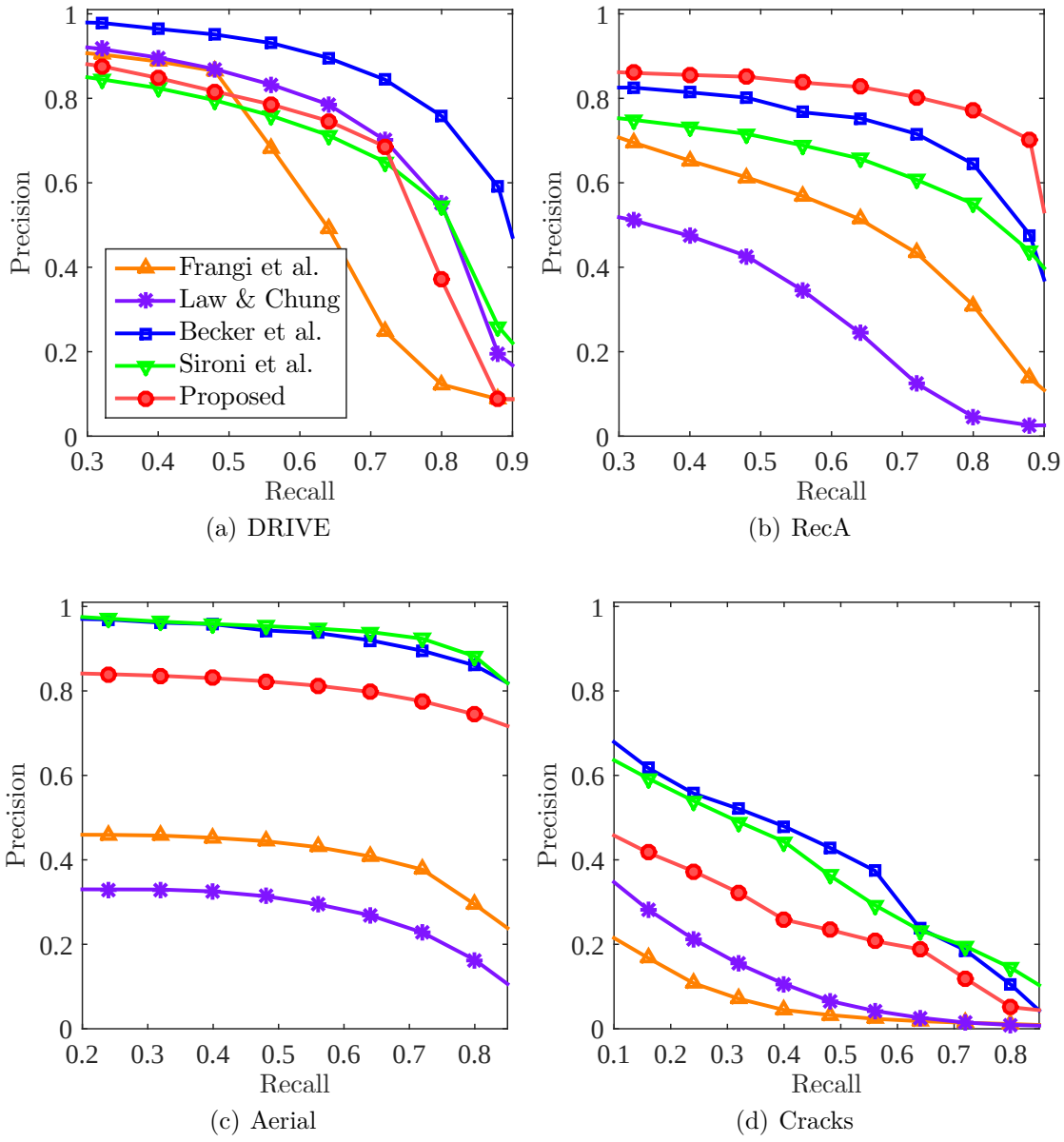
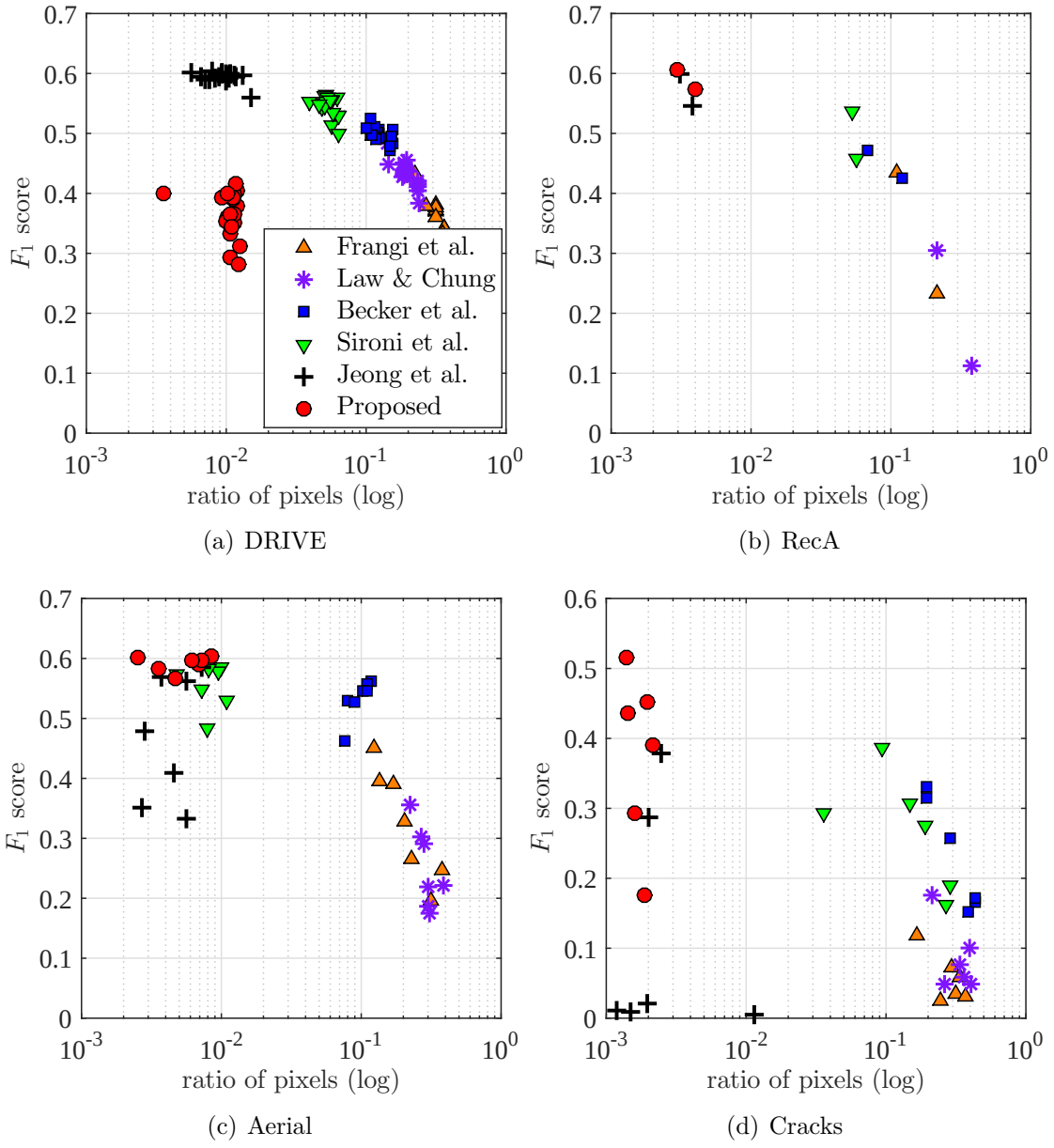


Figure 4.20: Precision-and-recall curve of the curvilinear structure segmentation algorithms for each dataset. We plot the curve by controlling threshold values for each algorithm: [Frangi 98], [Law 08], [Becker 13], [Sironi 14] and dissimilarity scores of the proposed algorithm.



(a) DRIVE (b) RecA

(c) Aerial (d) Cracks

Figure 4.21: We analyze the relationship between the average proportion of pixels to draw the curvilinear structure and the corresponding F_1 score for each dataset.

based on machine learning. Manual segmentation (ground truth) contains many errors around boundaries and minutiae components. Also, the number of training data employed in this work is relatively small due to the difficulties of making the accurate annotations. Specifically, since RecA dataset has only four training images, there is a high risk of over-fitting to the training set. It is remarkable that the proposed algorithm shows good performance for all datasets without over-fitting problem when compared to other learning based algorithms [Becker 13, Sironi 14].

4.7 Conclusions

This chapter proposes a new curvilinear structure reconstruction algorithm based on the ranking learning system and graph theory. The output rankings of the line segments correspond to the plausibility of the latent curvilinear structure. Using an optimal number of pixels, the proposed algorithm provides different levels of detail during reconstruction of the curvilinear structure. More precisely, we learn a ranking function based on SSVM with the proposed orientation-aware curvilinear feature descriptor. The weighted average of curvilinear feature groups improved the consistency of the output rankings. In this chapter, we define a novel graphical model that infers the curvilinear structure according to the topological importance. The proposed algorithm looks for remote vertices on the subgraph which is induced from the output rankings. Across the various types of datasets, our model shows good performances to reconstruct the latent curvilinear structure with a smaller number of pixels compared to the state-of-the-art algorithms. Furthermore, to our best knowledge, this is the first work to provide the topological importance of the curvilinear structure.

Conclusions and Perspectives

Conclusions and Perspectives

IN this dissertation, we proposed curvilinear structure reconstruction models which can interpret various types of natural images. This research topic let us ask a question. "How can we well represent curvilinear structure in general situations?" Geometrically speaking, curvilinear structures exhibit thin, elongated, and symmetric for a particular direction. Also, pixels on the curvilinear structure show slightly different intensity values compared to its surroundings. These are generally accepted assumptions in order to describe curvilinear structure in computer vision and image processing. Thus, oriented image gradient information and morphological filtering responses have been employed to extract such curvilinear features. In this thesis, we specifically assumed that the entire curvilinear structure can be separated into multiple line segments. We look for a set of line segments corresponding to the above mentioned characteristics of curvilinear structure. Each line segment is parameterized as its center pixel coordinate, length, and orientation. We further studied recently developed computer vision techniques, *e.g.,* stochastic modeling, structured information prediction, and graph theory.

In Chapter 2, we first reviewed the existing curvilinear structure reconstruction models to analyze blood vessels, biological images, facial wrinkles, road networks, and defects in the asphalt. Most of the previous works have been designed for specific applications due to the lack of evidence to detect curvilinear structure using local image features and complex topological behaviors depending on the applications. We found a common description of the curvilinear structure based on its geophysical characteristics, such that the curvilinear structure exhibits thin, elongated, and symmetrical structure on images. We obtained corresponding image features via rotated gradient information and morphological image filtering. We also realized that geometric priors are essential to improve the performances by inferring missing information.

We implemented two different curvilinear structure reconstruction models with stochastic model in Chapter 3 and machine learning in Chapter 4. Chapter 3 began to propose a probability density of line segments for given image data. To maximize the proposed density, we employed RJMCMC sampler with delayed rejection. The proposed stochastic model requires a small number of hyperparameters to decide relevant

importance between geometric constraints. Instead of estimating the exact values of these hyperparameters, we simulated several Markov chains to obtain line hypotheses. An aggregation algorithm of these line hypotheses is proposed to reduce sampling space and to find the most promising parameter set.

Chapter 4 proposed a new curvilinear structure representation method based on graph theory which exploits output ranking of the line segments. The proposed curvilinear structure inference algorithm visualized topological importance according to the length of the structure. We learned a ranking function to predict the correspondence between line segment and the latent curvilinear structure. The proposed algorithm efficiently reconstructed curvilinear structure with the minimum number of pixels. For each chapter, we compared the quantitative and qualitative performances of the proposed algorithm and of the competing methods.

The proposed algorithms aim to find an optimal set of line segments to represent the latent curvilinear structure. The same curvilinear features (steerable gradient profiles and morphological profiles) are employed in the proposed models. In Chapter 3, stochastic model for curvilinear structure is proposed to localize a set of line segments onto the latent curvilinear structure. It is useful for application to exploit information regarding distribution of the line segments. For instance, to evaluate facial wrinkle condition, the density of the line segments can be a criterion. In Chapter 4, we propose a curvilinear structure reconstruction algorithm based on ranking learning system and graph traversal algorithm. The later proposed algorithm has the merit of preserving topological feature of the curvilinear structure. Thus, it shows robust reconstruction results without discontinuities, where the input image consists of noisy background textures.

Perspectives

Our stochastic model proposed in Chapter 3 often showed disconnected line segments at junctions (see Figure B.2). It is because the prior model is designed to reject large curvatures in the configuration. We can remedy this drawback by employing another layer to represent the curvilinear structure in a different form *e.g.,* Bezier curve networks [Bama 15] or a sequence of pixels. We define a transform function, which is invertible, from a group of line segments to a desired representation. We assume a strong connectivity of the pixels belonging to the same group. To implement this model, we shall design a new transition kernel for MCMC sampler which can jump between different representation layers in a reversible manner.

For our second model regarding machine learning, we can improve the feature descriptor to take into account the symmetric shape. This term will contribute to high-

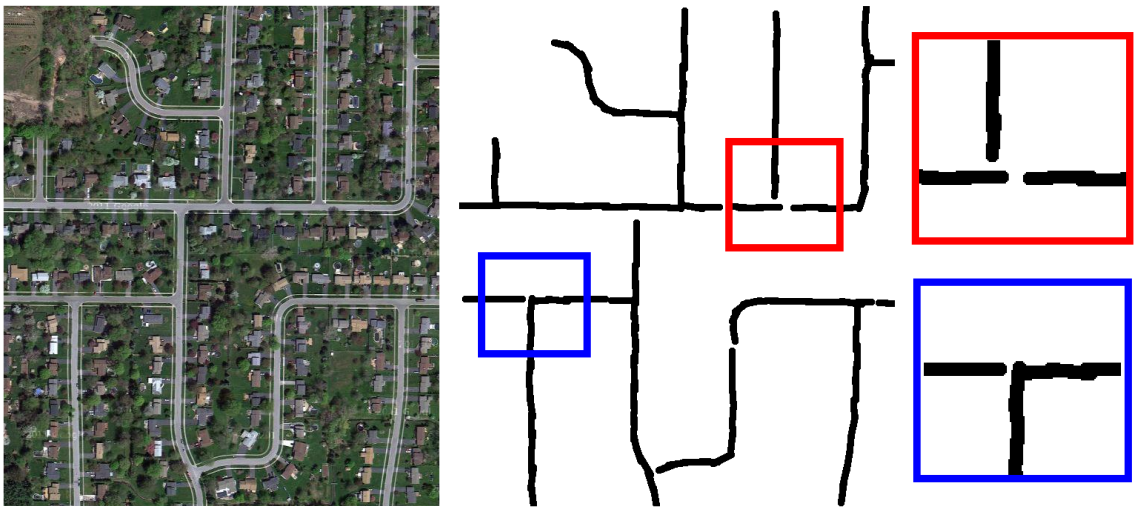


Figure 5.22: Failure case of our stochastic model proposed in Chapter 3. Specifically, prior constraints penalize a large curvature of the local configuration, so that yield conflict at junction areas.

light output rankings at the center of the linear structure. The idea is inspired by the work of [Sironi 14] who exploits distance transform to identify centerlines. Moreover, we consider training non-line shape templates such as bifurcation patterns [Azzopardi 11] in order to localize junctions.

In this thesis, each line segment is parameterized as its center pixel coordinate, length, and orientation. Missing the parameter to describe thickness, the proposed algorithms show performance drops for detection of width-varying curvilinear structures *i.e.,* blood vessels. To overcome this issue, we can add a thickness parameter to describe line segment. We also consider extending the proposed models with hierarchical grouping.

Certainly, the idea involving machine learning technique (employed in Chapter 4) with MCMC optimization (proposed in Chapter 3) is very interesting. We expect that the hyperparameters can be learned and updated at the each iteration of MCMC optimization (coupled with simulated annealing). Specifically, Wick *et al.* [Wick 11] proposed the ranking learning algorithm called *SampleRank* which can be combined with MCMC optimization. However, our primitive attempt to apply this method into the proposed cost function has failed due to the following reasons: First, the cost function of the stochastic model is non-convex because of the overlapping penalty term (3.16) that prevents congestion of the line segments in the local configuration. Second, since the groundtruth map is provided by a binary segmentation map, we do not have

information regarding pairwise relation of line segments. Without this information, it was difficult to define a loss function which measures the quality of the estimated parameter for the cost function.

We can also consider to speed up the proposed algorithms by employing parallel programming. Specifically, the optimization process of our stochastic model supports the parallel implementation. The computational time can be further reduced by applying a parallel Monte Carlo sampler [Verdié 14].

Appendix

Appendix A

Data Sets

THIS chapter describes the datasets used for the experiments conducted in this thesis. We evaluate the proposed curvilinear structure reconstruction models on various types of images, such as microscopic images of RecA-DNA filaments; retinal images of blood vessels; facial wrinkles; satellite images of road networks; and defects in the asphalt. The details regarding each dataset with the representative images are presented below.

A.1 DNA: Microscopic images of RecA-DNA filaments

In biological chemistry, negative stain of electron microscopy visualizes protein complexes [Jain 95]. We collect electron microscopic images of RecA proteins which contain filament structure (see Figure A.1).

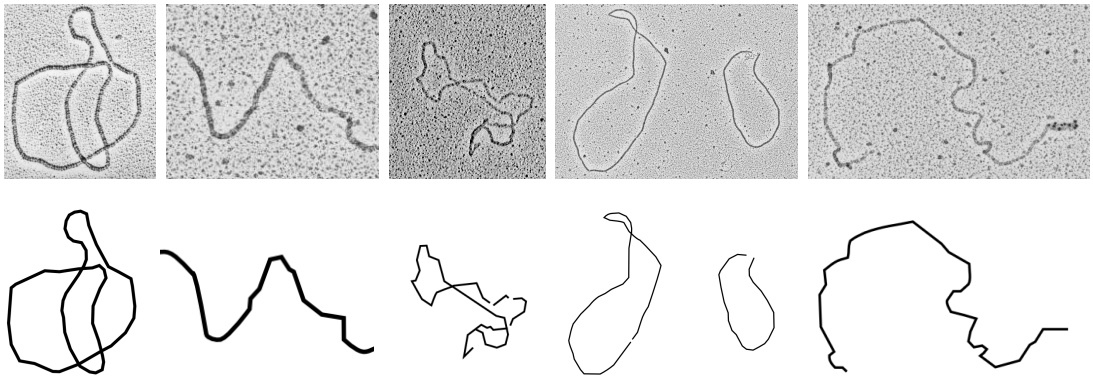


Figure A.1: Example images of DNA dataset

A.2 Drive: Retinal images of blood vessels

This dataset [Staal 04] contains 40 retinal images with binary masks and ground truth images, which are manually annotated by two ophthalmologists. Specifically, the dataset provides 20 images for the training phase and 20 images for the testing phase. Blood vessels corresponding to curvilinear structure are darker than backgrounds. Due to the spherical shape of eyeball, luminance of the retinal images is unevenly highlighted. In Figure A.2, we provide several images of the Drive dataset and the corresponding binary masks and ground truth images. More detailed description of this dataset is availabe at the website[1] of the authors [Staal 04].

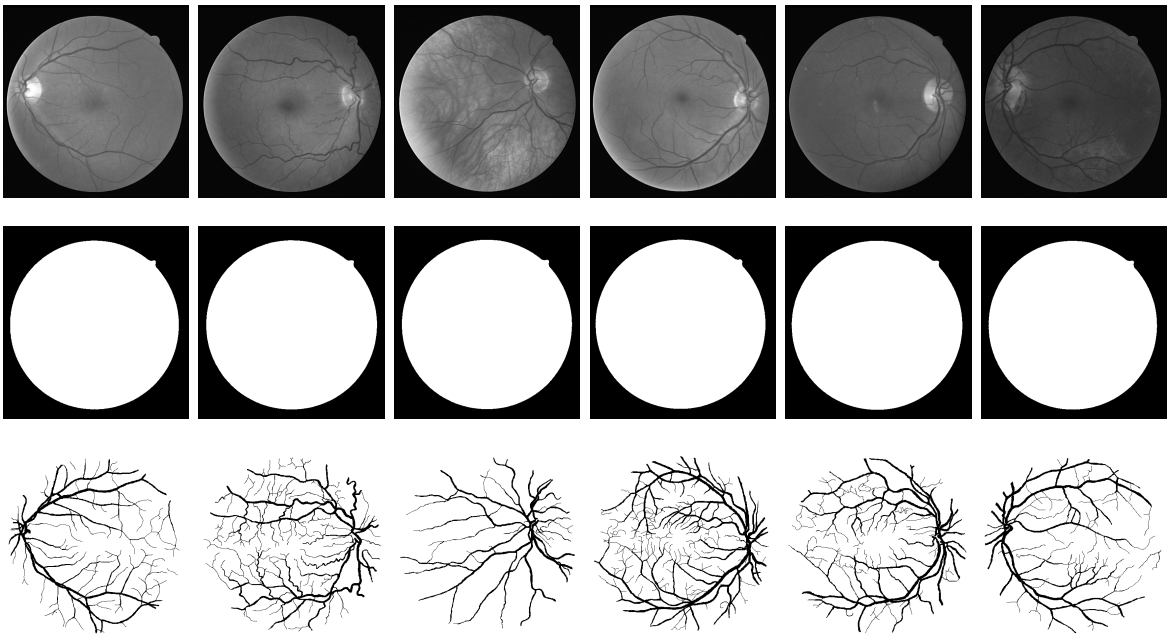


Figure A.2: Example images of Drive dataset

[1] http://www.isi.uu.nl/Research/Databases/DRIVE/

A.3 Wrinkle: Facial wrinkles

We collect facial wrinkle images of well known people on the Internet, such as actors and politicians. We also tested our methods on the facial skin images which are collected by world a leading dermatology company; however, these images cannot be presented in this thesis due to the permission issues. Facial wrinkles can be occluded by hairs and accessories or covered by makeup. Thus, we mainly test the algorithms to detect the wrinkles on the forehead part, where the corresponding areas are manually labeled before applying the algorithms. Figure A.3 presents some images of WRINKLE dataset, the corresponding mask, and ground truth, respectively.

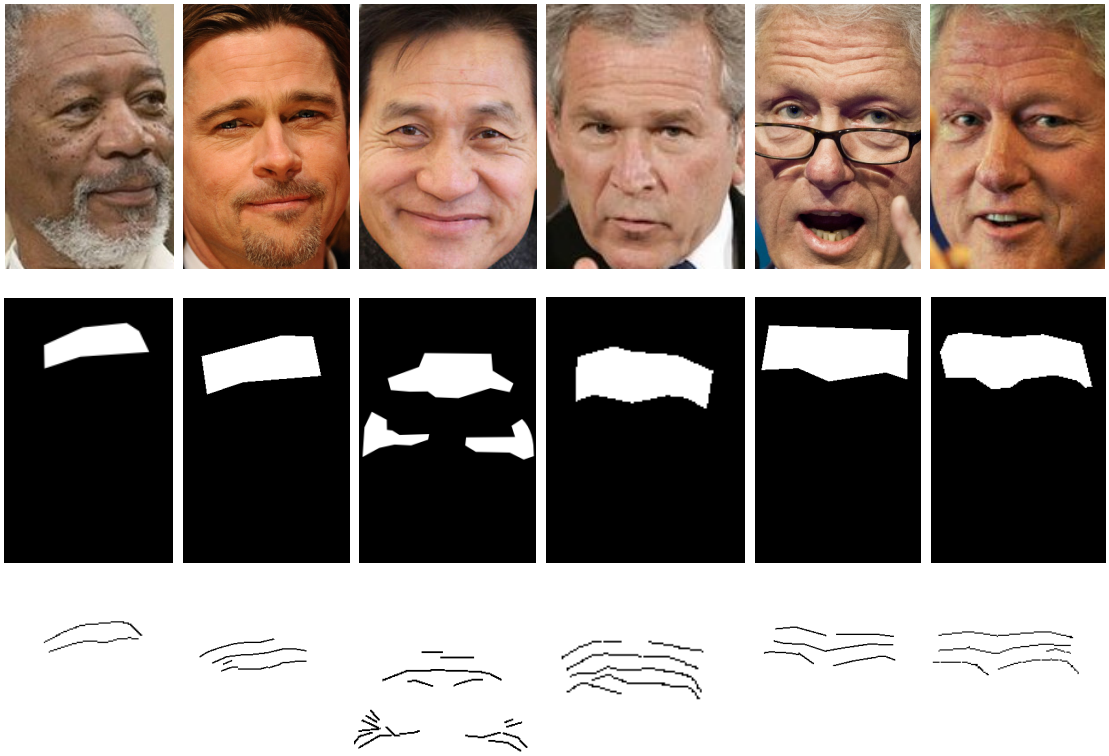


Figure A.3: Example images of WRINKLE dataset

A.4 Aerial: Satellite images of road networks

We test satellite images containing road networks to evaluate the curvilinear structure reconstruction algorithms. The dataset contains 14 remote sensing images of road networks and divided as 7 images for training and 7 images for the testing, respectively. Curvilinear structures associated with road network are partially occluded by the trees or cars on the road. This dataset is courtesy of the authors of [Sironi 14] and available at their website[2]. Some images of this dataset are provided in Figure A.4.

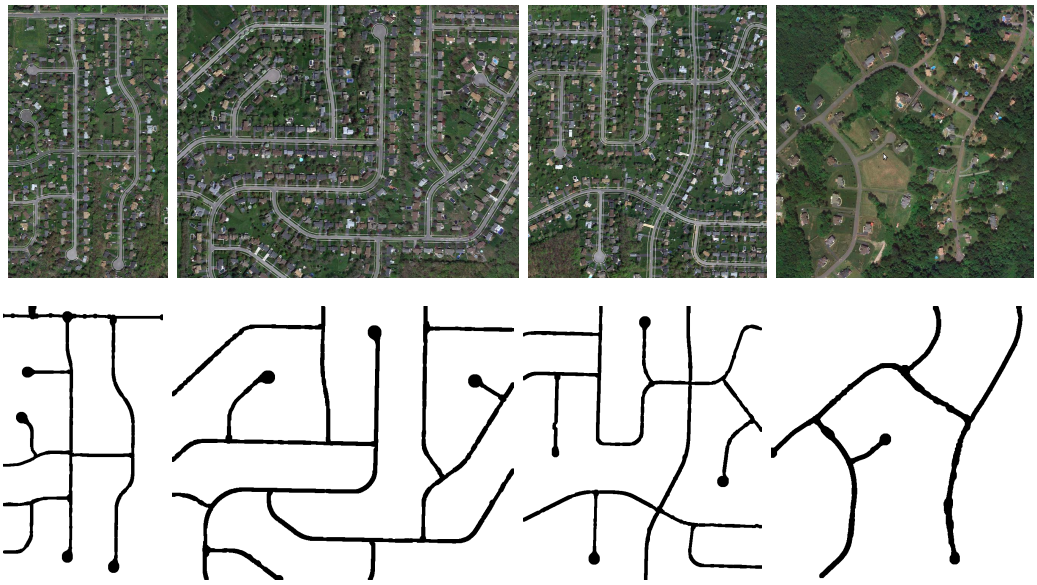


Figure A.4: Example images of Aerial dataset

[2]http://cvlab.epfl.ch/software/centerline-detection

A.5 Crack: Defects in the asphalt

Images regarding road cracks contain thin and elongated structures in the asphalt surfaces (see Figure A.5). The dataset provides the real and synthetic images of various resolutions. We select 12 real images in the dataset, which have the same resolution (768×512). We use 6 images to train, and different images 6 to test the algorithms. Please note that images are under copyright of the IFSTTAR institute, France. We obtain this experimental data courtesy of the authors [Chambon 10].

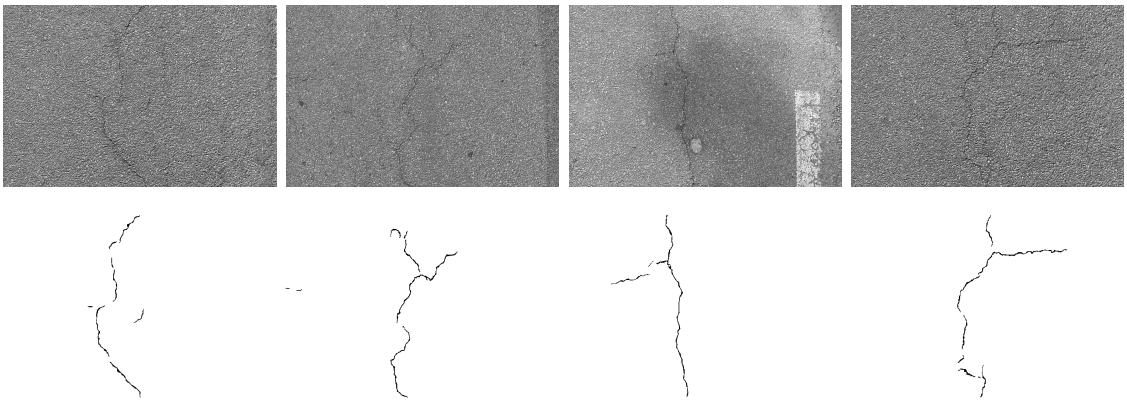


Figure A.5: Example images of CRACK dataset

Appendix B

Résumé étoffé de la thèse en Français

B.1 Introduction

La recherche et les industries liées à la vision par ordinateur ont eu un développement rapide ces dernières années en raison de la fiabilité des capteurs et du développements de la puissance de calcul, cela en analysant différents types d'images. De l'imagerie microscopique à l'imagerie satellitaire, nous observons que des structures curvilignes existent sous différentes formes. Il est donc nécessaire de détecter les structures curvilignes latentes afin d'améliorer la visibilité des objets recherchés, d'extraire des informations biologiques, et d'analyser l'information géographique. Alors que l'homme peut intuitivement distinguer ces structures curvilignes latentes, la détection automatique de ces structures curvilignes est difficile à ce jour. Les difficultés sont dues au fait que ces structures curvilignes sont présentées sous différentes topologies. Ainsi, il est impossible de définir ces structures curvilignes complexes avec seulement quelques paramètres. Selon l'environnement de l'acquisition, les structures curvilignes peuvent être cachées sous les textures de fond avec un faible contraste. Par conséquent, la représentation des données et de nombreux algorithmes de vision par ordinateur, pour détecter les structures curvilignes, ont été développés pour des applications spécifiques. La figure B.1 montre des applications variées pour l'utilisation des structures curvilignes.

En vision bas niveau, des chercheurs [Freeman 91, Perona 95, Jacob 04] ont développé des filtres de convolution pour extraire les caractéristiques curvilignes de la texture du fond. L'idée principale derrière cette conception de filtre est de créer des modèles de forme de ligne pour extraire des informations de gradients orientés localement. Frangi *et al.* analysent les valeurs propres de la matrice de hessien pour une image

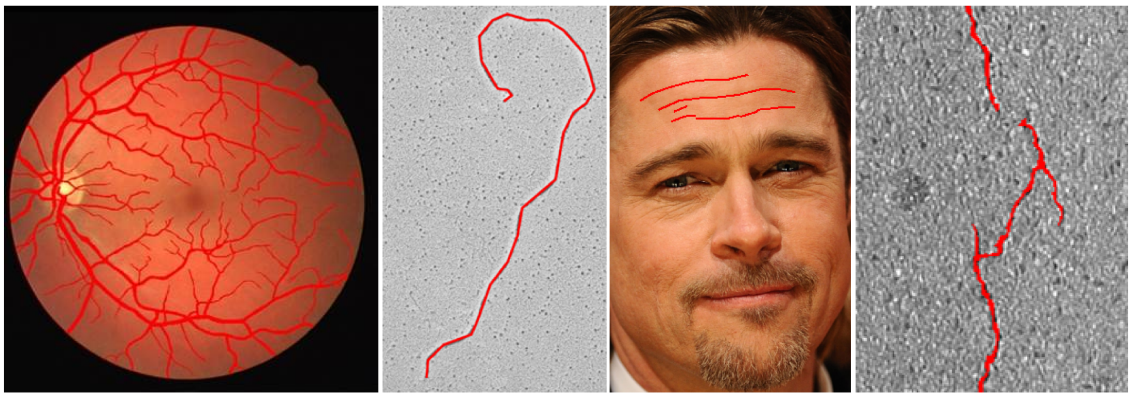


Figure B.1: Des structures curvilignes sont observées avec différentes caractéristiques topologiques et diverses textures de fond. De gauche à droite des images montrent une structure curviligne latente de vaisseaux sanguins, de filaments d'ADN, de rides du visage et de fissures sur une route.

donnée afin d'obtenir les directions principales de la structure locale [Frangi 98]. Le flux optimal orienté (Optimally Oriented Flux en anglais, OOF) mesure la quantité de flux de gradient sortant pour trouver les structures curvilignes [Law 08]. L'opérateur morphologique recueille des pixels en fonction de la similitude structurelle sur le chemin allongé [Talbot 07]. Toutefois, en raison de l'absence d'interprétation de forme, ces méthodes fondées sur les caractéristiques locales d'image sont insuffisantes pour reconstruire la structure curviligne sous-jacente. Par ailleurs, des modèles graphiques tels que [González 10, Türetken 13b] définissent des contraintes géométriques dans une configuration locale en minimisant globalement leurs fonctions de coût. Plus précisément, les algorithmes fondés sur des graphes sont initialisés par des points correspondant à des structures curvilignes latentes, puis définissent un trajet qui relie ces points avec un a priori géométrique pour fournir des formes plausibles. Les propriétés géométriques de réseau de lignes sont impliqués comme termes de contraintes quand un problème d'optimisation de l'énergie est formulé. Des modèles stochastiques [Lacoste 05, Jeong 15a] reconstruisent les structures curvilignes en échantillonnant plusieurs segments de ligne afin de maximiser une probabilité a posteriori de données d'image fournie. Des concepts allant de la similarité à la représentation graphique sur des a priori géométriques sont considérés pour définir la connectivité et la courbure des segments de ligne. Récemment, des algorithmes d'apprentissage automatique ont été proposés pour détecter la structure curviligne. Becker *et al.* [Becker 13] ont appliqué un algorithme pour obtenir un ensemble optimal de bancs de filtres de convolution. Sironi *et al.* [Sironi 14] ont développé un modèle de régression pour estimer l'échelle (largeur) des structures curvilignes et localiser leurs axes.

Bien que les algorithmes de [Tu 06, Arbeáez 11] détectent des courbes et des lignes, l'objectif est tout à fait différent des techniques de reconstruction des structures curvilignes. Les algorithmes de contour mentionnés ci-dessus cherchent des lignes de contour fermé pour diviser une image en régions significatives. Par conséquent, des fonctions de coût exploitent les signaux globaux de texture en ce que les contours sont associés à des bords saillants près des limites de l'objet. A l'opposé, nous recherchons plusieurs structures curvilignes, qui ne sont pas nécessairement fermées et sont latentes dans une texture homogène. En comparaison avec le contour, les structures curvilignes sont estimées via de subtiles caractéristiques d'images locales. Les similitudes internes de la structure et la conception précise de la forme sont essentiels pour résoudre notre problème.

B.2 Contributions

Dans cette thèse, nous avons comme objectif ambitieux de proposer un cadre général qui peut analyser différents types de structures curvilignes variées. Nous supposons que la structure curviligne peut être décomposée en un ensemble de segments de ligne. Nous cherchons donc un ensemble optimal de segments de ligne qui correspond à la structure curviligne latente. Nous avons proposé deux modèles de reconstruction de la structure curviligne fondés respectivement sur un modèle stochastique avec une technique d'échantillonnage (Chapitre 3) et sur un classifieur à partir d'un système d'apprentissage avec une représentation graphique (Chapitre 4). Pour le modèle stochastique, nous proposons une probabilité a posteriori de segments de ligne pour l'image donnée, et nous optimisons la répartition proposée par un echautillonnage de type Markov chain Monte Carlo (MCMC). Notre travail contribue spécialement à réduire la dépendance pour les hyperparamètres du processus stochastique. À cette fin, nous simulons simultanément plusieurs chaînes de Markov avec différents paramètres et des hypothèses multiples de lignes agrégées. En outre, nous apprenons une fonction de classement qui mesure la comparabilité des segments de ligne donnés et les structures curvilignes latentes. Une méthode de représentation graphique utilisant les classements de sortie des segments de ligne est proposée pour montrer l'importance de la structure topologique curviligne sous-jacente. Nous testons différents types d'ensembles de données, et fournissons des résultats expérimentaux dans chaque chapitre.

B.2.1 Chapitre 2 : Modélisation de la structure curviligne : présentation générale

Dans le chapitre 2, nous fournissons des descripteurs de structure curviligne fondés sur le gradient de l'image et des profils morphologiques. Nous percevons la structure curviligne latente sur la base de l'incohérence des textures de fond et des caractéristiques géométriques. En d'autres termes, une séquence de pixels correspondant à la structure curviligne a des valeurs d'intensité différentes comparé à son environnement, et exhibe une forme fine et allongée. Le gradient de l'image est calculé par rapport à diverses orientations pour détecter des fonctionnalités orientées localement. En outre, nous employons des profils morphologiques pour trouver des pixels représentant les différéntes textures du fond sur un chemin allongé. Dans ce chapitre, nous examinons aussi les travaux antérieurs sur la structure des modèles de reconstruction curviligne tels que la segmentation, la représentation à base de graphes, la modélisation stochastique, et l'apprentissage automatique (machine learning en anglais).

B.2.2 Chapitre 3 : Modèle stochastique

Dans le chapitre 3, nous étudions un modèle stochastique pour la reconstruction de la structure curviligne. Nous proposons une densité de probabilité de segments de ligne pour une image donnée, où cette densité de probabilité est constituée de la vraisemblance et de l'a priori. Pour la probabilité au terme d'attache aux données (vraisemblance), nous exploitons le gradient de l'image pivotée et les réponses des filtres morphologiques pour estimer la présence d'une structure curviligne. Nous approchons les statistiques de caractéristiques curvilignes en utilisant une distribution gaussienne avec un SVM (Support Vector Machine). Nous limitons aussi des comportements géométriques locaux de segments de ligne. Nous encourageons la bonne connexion de segments de ligne, alors que nous rejetons l'encombrement de segments de ligne. Pour optimiser la densité proposée, nous employons un algorithme de type MCMC à saut réversible (RJMCMC) avec un rejet retardé. La principale contribution de ce chapitre est d'éliminer la dépendance de la performance de la sélection des hyperparamètres en utilisant l'optimisation stochastique. Plusieurs chaînes de Markov sont simulées avec différents vecteurs d'hyperparamètres afin d'obtenir différentes hypothèses de ligne. Puis nous agrégeons ces hypothèses afin de réduire l'espace d'échantillonnage. Nous fournissons des expériences pour comparer les résultats de reconstruction de la structure curviligne obtenue par l'algorithme proposée et des algorithmes concurrents de l'état de l'Art.

B.2.3 Chapitre 4 : Inférence de la structure curviligne

Le chapitre 4 traite d'un modèle d'inférence de structure curviligne qui quantifie un score en fonction de l'entrée et de la sortie des informations de paires de segments de ligne. Pour la classification, les méthodes antérieures définissent globalement un seuil pour supprimer des pixels qui montrent une faible probabilité d'appartenir à la structure curviligne. Au cours de ce processus, la plupart des travaux précédents ignorent implicitement les informations corrélées des pixels situés sur la structure curviligne. Nous visons à apprendre une fonction de classement des segments de ligne pour des structures curvilignes données. Le classement de sortie est en mesure d'organiser les segments de ligne dans l'ordre de correspondance avec les structures curvilignes latentes. Plus précisément, un SVM Structuré (SSVM) est utilisé pour déduire les classements de sortie. Nous visualisons ensuite les ordres topologiques de la structure curviligne. Nous étudions la théorie des graphes pour reconstruire le trajet curviligne qui relie les plus reculés des pixels sur la structure curviligne latente. L'algorithme proposé ajoute progressivement les branches les plus fines dans la structure principale jusqu'à ce qu'il trouve la longueur minimale de la structure. Nous montrons des résultats expérimentaux sur divers ensembles de données pour illustrer l'efficacité de l'algorithme proposé.

B.3 Conclusions

Dans cette thèse, nous avons proposé des modèles de reconstruction de structures curvilignes qui peuvent interpréter différents types d'images naturelles. Posons une question relative à notre sujet de recherche : "Comment pouvons-nous bien représenter la structure curviligne dans des situations générales?" Pour répondre à cette question, nous supposons que la structure curviligne peut être représentée comme une collection de nombreux segments de ligne. Nous avons étudié davantage des techniques récemment développées en vision pas ordinateur, par exemple, la modélisation stochastique, la prédiction de l'information structurée, et la théorie des graphes.

Dans le chapitre 2, nous avons d'abord examiné les modèles de reconstruction de structures curvilignes existants pour analyser des vaisseaux sanguins, des images biologiques, des rides du visage, des réseaux routiers et des défauts dans l'asphalte. La plupart des travaux précédents ont été conçus pour des applications spécifiques en raison de l'absence de preuve pour détecter des structures linéaires curvilignes utilisant des caractéristiques de l'image et des comportements locaux topologiques complexes en fonction des applications. Nous avons proposé une description commune de la structure curviligne basée sur des caractéristiques géométriques, telles que la finesse, l'allongement et symétrie sur les images. Nous avons obtenu des caractéristiques de

l'image correspondante par des informations de gradient orienté et par filtrage morphologique d'image. Nous avons également réalisé que des a priori géométriques sont essentiels pour améliorer les performances en inférant des informations manquantes.

Nous avons proposé deux modèles de reconstruction de structures curvilignes différentes : un modèle stochastique dans le chapitre 3 et un modèle fondé sur l'apprentissage automatique dans le chapitre 4. Le chapitre 3 a commencé par proposer une densité de probabilité de segments de ligne d'images données. Afin de maximiser la densité proposée, nous avons utilisé l'échantillonneur RJMCMC avec un rejet retardé. Le modèle stochastique proposé nécessite le choix d'hyperparamètres pour décider de l'importance relative entre les contraintes géométriques. Au lieu d'estimer les valeurs exactes de ces hyperparamétres, nous avons simulé plusieurs chaînes de Markov pour obtenir des hypothèses de ligne. Un algorithme d'agrégation de ces hypothèses de ligne est proposé pour réduire l'espace d'échantillonnage et trouver l'ensemble des paramètres les plus prometteurs.

Le chapitre 4 a proposé une nouvelle méthode de représentation de la structure curviligne fondée sur la théorie des graphes qui exploite le classement de sortie des segments de ligne. L'algorithme d'inférence de structures curvilignes proposé visualise l'importance topologique en fonction de la longueur de la structure. Nous avons appris une fonction de classement afin de prédire la correspondance entre les segments de ligne et la structure curviligne latente. L'algorithme proposé reconstruit efficacement la structure curviligne avec un nombre minimum de pixels utilisés. Pour chaque chapitre, nous avons comparé les performances quantitatives et qualitatives de l'algorithme proposé avec des méthodes concurrentes de l'état de l'Art.

B.4 Perspectives

Notre modèle stochastique proposé au chapitre 3 a souvent trouvé des segments déconnectés de jonctions en T (voir Figure B.2). Car le modéle est conçu pour rejeter de grandes courbures dans la configuration. Nous pouvons remédier à cet inconvénient en utilisant une autre couche pour représenter la structure curviligne sous une forme différente, par exemple, des réseaux de courbes de Bézier [Bama 15] ou une séquence de pixels. On pourrait définir une fonction, qui soit inversible, transformée à partir d'un groupe de segments de ligne en une représentation souhaitée. Nous supposons une forte connectivité des pixels appartenant au même groupe. Pour mettre en œuvre ce modèle, nous devons concevoir un nouveau noyau de transition pour l'échantillonneur MCMC à saut réversible.

En ce qui concerne notre second modèle, fondé sur l'apprentissage automatique, nous pourrions améliorer le descripteur de fonction pour tenir compte de la forme

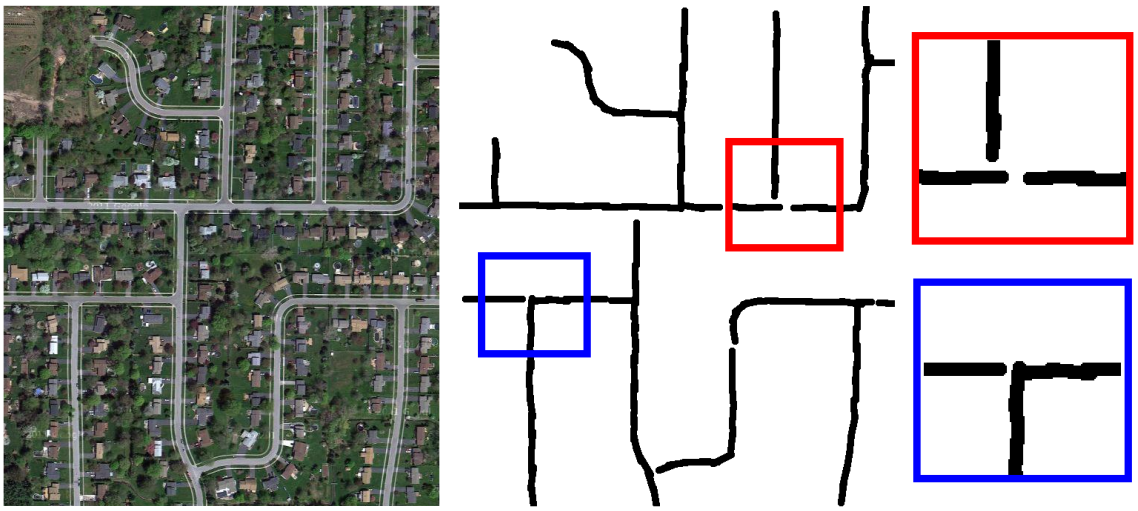


Figure B.2: Cas de défaillance du modèle stochastique proposé au chapitre 3. Plus précisément, les contraintes antérieures pénalisent une grande courbure de la configuration locale, pour reduire les conflicts dans les zones de jointure.

symétrique. Ce terme contribuerait à mettre en évidence les classements de sortie au centre de la structure linéaire. L'idée est inspirée par le travail de [Sironi 14] qui exploite la transformée de distance pour identifier les axes. Aussi, nous pourrions considérer des structures non-curvilignes telles que des bifurcation en V ou en T par exemple, afin de localiser les discontinuités [Azzopardi 11].

Nous pourrions aussi accélérer les algorithmes proposés en employant la programmation parallèle. Plus précisément, le processus d'optimisation de notre modèle stochastique est parallelisable. Le temps de calcul peut être encore réduit par l'implémentation parallèle de l'échantillonneur RJMCMC [Verdié 14].

Bibliography

[Al-Diri 09] B. Al-Diri, A. Hunter & D. Steel. *An active contour model for segmenting and measuring retinal vessels.* IEEE TMI, vol. 28, no. 9, pages 1488–1497, 2009.

[Arandjelović 11] R. Arandjelović & A. Zisserman. *Smooth Object Retrieval using a Bag of Boundaries.* In ICCV, pages 374–382, 2011.

[Arbeáez 11] P. Arbeáez, M. Maire, C. Fowlkes & J. Malik. *Contour detection and hierarchical image segmentation.* IEEE TPAMI, vol. 33, no. 5, pages 898–916, 2011.

[Arbelaez 10] P. Arbelaez, M. Maire, C. Fowlkes & J. Malik. *Contour detection and hierarchical image segmentation.* IEEE TPAMI, vol. 33, no. 5, pages 898–916, 2010.

[Azzopardi 11] G. Azzopardi & N. Petkov. *Detection of retinal vascular bifurcations by trainable V4-like filters.* In CAIP, pages 451–459, 2011.

[Bama 15] K. Bama, T. Szirányi, M. Borda & O. Lavialle. *Marked point processes for enhancing seismic fault patterns.* Journal of Applied Geophsics, vol. 118, pages 115–123, 2015.

[Batool 12] N. Batool & R. Chellappa. *Modeling and detection of wrinkles in aging human faces using marked point processes.* In ECCV Ws/Demos, pages 178–188, 2012.

[Batool 13] N. Batool, S. Taheri & R. Chellappa. *Assessment of facial wrinkles as a soft biometrics.* In FG, pages 1–7, 2013.

[Becker 13] C. Becker, R. Rigamonti, V. Lepetit & P. Fua. *Supervised feature learning for curvilinear structure segmentation.* In MICCAI, pages 526–533, 2013.

[Bertelli 11] L. Bertelli, T. Yu, D. Vu & B. Gokturk. *Kernelized structural SVM learning for supervised object segmentation.* In CVPR, pages 2153–2160, 2011.

[Besag 86] J. Besag. *On the statistical analysis of dirty pictures.* Journal of Royal Statistical Society B, vol. 48, no. 3, pages 259–302, 1986.

[Borassi 15] M. Borassi, P. Crescenzi, M. Habib, W. A. Kosters, A. Marino & F. W. Takes. *Fast diameter and radius BFS-based computation in (weakly connected) real-world graphs: With an application to the six degrees of separation games.* Theor. Comput. Sci., vol. 586, pages 59–80, 2015.

[Bouix 05] S. Bouix, K. Siddiqi & A. Tannenbaum. *Flux driven automatic centerline extraction.* Med Image Anal, vol. 9, no. 3, pages 209–221, 2005.

[Boyd 04] S. Boyd & L. Vandenberghe. Convex optimization. Cambridge University Press, New York, NY, USA, 2004.

[Boykov 04] Y. Boykov & V. Kolmogorov. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.* IEEE TPAMI, vol. 26, no. 9, pages 1124–1137, 2004.

[Brown 11] K. M. Brown, G. Barrionuevo, A. J. Canty, V. De Paola, J. A. Hirsch, G. S. X. E. Jefferis, J. Lu, M. Snippe, I. Sugihara & G. A. Ascoli. *The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions.* Neuroinformatics, vol. 9, no. 2–3, pages 143–157, 2011.

[Canny 86] J. Canny. *A computational approach to edge detection.* IEEE TPAMI, vol. 8, no. 6, pages 679–698, 1986.

[Celeux 96] G. Celeux, D. Chauveau & J. Diebolt. *Stochastic versions of the EM algorithm: an experimental study in the mixture case.* J. Statist. Comput. Simulation, vol. 55, no. 4, pages 287–314, 1996.

[Chambon 10] S. Chambon, C. Gourraud, J.-M. Moliard & P. Nicolle. *Road crack extraction with adapted filtering and Markov model-based segmentation.* In VISAPP(2), pages 81–90, 2010.

[Chambon 11] S. Chambon & J.-M. Moliard. *Automatic road pavement assessment with image processing: review and comparison.* International Journal of Geophysics, vol. 2011, pages 1–20, 2011.

[Chatelain 12] F. Chatelain, X. Descombes, F. Lafarge, C. Lantuejoul, C. Mallet, R. Minlos, M. Schmitt, M. Sigelle, R. Stoica & E. Zhizhina. Stochastic geometry for image analysis. Wiley-ISTE, 2012.

[Chaudhuri 89] S. Chaudhuri, S. Chatterjee, N. Katz & M. Nelsonand M. Goldbaum. *Detection of blood vessels in retinal images using two-dimensional matched filters.* IEEE TMI, vol. 8, no. 3, pages 263–269, 1989.

[Corneil 01] D. G. Corneil, F. F. Dragan, M. Habib & C. Paul. *Diameter determination on restricted graph families.* Discrete Applied Mathematics, vol. 113, no. 2–3, pages 143–166, 2001.

[Cula 05] O. G. Cula, K. J. Dana, F. P. Murphy & B. K. Rao. *Skin texture modeling.* IJCV, vol. 62, no. 1–2, pages 97–119, 2005.

[Das 11] S. Das, T. T. Mirnalinee & K. Varghese. *Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images.* IEEE TGRS, vol. 49, no. 10, pages 3906–3931, 2011.

[Descombes 02] X. Descombes & J. Zerubia. *Marked point process in image analysis.* IEEE SPM, vol. 19, no. 5, pages 77–84, 2002.

[Dijkstra 59] E. W. Dijkstra. *A note on two problems in connexion with graph.* Numer. Math., vol. 1, no. 1, pages 269–271, 1959.

[Fischler 81] M. A. Fischler, J. M. Tenenbaum & H. C. Wolf. *Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique.* Computer Graphics and Image Processing, vol. 15, no. 3, pages 201–223, 1981.

[Frangi 98] A. F. Frangi, W. J. Niessen, K. L. Vincken & M. A. Viergever. *Multiscale vessel enhancement filtering.* In MICCAI, pages 130–137, 1998.

[Freeman 91] W. T. Freeman & E. H. Adelson. *The design and use of steerable filters.* IEEE TPAMI, vol. 13, no. 9, pages 891–906, 1991.

[Fu 10] Y. Fu, G. Guo & T. Huang. *Age synthesis and estimation via faces: a survey.* IEEE TPAMI, vol. 32, no. 11, pages 1955–1976, 2010.

[Gamal-Eldin 10] A. Gamal-Eldin, X. Descombes, G. Charpiat & J. Zerubia. *Multiple birth and cut algorithm for point process optimization.* In SITIS, 2010.

[Geman 84] S. Geman & D. Geman. *Stochastic relaxation, Gibbs distribution and the bayesian restoration of images*. IEEE TPAMI, vol. 6, no. 6, pages 721–741, 1984.

[Geman 90] S. Geman, D. Geman, C. Grafiigne & P. Dong. *Boundary detection by constrained optimization*. IEEE TPAMI, vol. 12, no. 7, pages 609–628, 1990.

[Geman 96] D. Geman & B. Jedynak. *An active testing model for tracking roads in satellite images*. IEEE TPAMI, vol. 18, no. 1, pages 1–14, 1996.

[Gibbs 02] J. W. Gibbs. Elementary principles in statistical mechanics. Charles Scribner's Sons, New York, NY, USA, 1902.

[Gilks 95] W. R. Gilks, S. Richardson & D. Spiegelhalter. Markov chain Monte Carlo in practice. Chapman & Hall/CRC, 1995.

[González 10] G. González, E. Türetken, F. Fleuret & P. Fua. *Delineating trees in noisy 2D images and 3D image-stacks*. In CVPR, pages 2799–2806, 2010.

[Green 95] P. J. Green. *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*. Biometrika, vol. 82, no. 4, pages 771–732, 1995.

[Green 01] P. J. Green & A. Mira. *Delayed rejection in reversible jump Metropolis-Hastings*. Biometrika, vol. 88, no. 4, pages 1035–1053, 2001.

[Grigorescu 02] S. E. Grigorescu, N. Petkov & P. Kruizinga. *Comparison of texture features based on Gabor filters*. IEEE TIP, vol. 11, no. 10, pages 1160–1167, 2002.

[Hamming 50] R. W. Hamming. *Error detecting and error correcting codes*. Bell System Technical Journal, vol. 29, no. 2, pages 147–160, 1950.

[Hastie 09] T. Hastie, R. Tibshirani & J. Friedman. The elements of statistical learning. Springer Series in Statistics, Springer New York, NY, USA, second edition, 2009.

[Hastings 70] W. K. Hastings. *Monte Carlo sampling methods using Markov chains and their applications*. Biometrika, vol. 57, no. 1, pages 97–109, 1970.

[Hu 07] J. Hu, A. Razdan, J. C. Femiani, M. Cui & P. Wonka. *Road network extraction and intersection detection from aerial images by tracking road footprints.* IEEE TGRS, vol. 45, no. 12, pages 4144–4157, 2007.

[Iyer 05] S. Iyer & S. Sinha. *A robust approach for automatic detection and segmentation of cracks in underground pipeline images.* Image and Vision Computing, vol. 23, no. 10, pages 921–933, 2005.

[Jaccard 12] P. Jaccard. *The distribution of the flora in the alpine zone.* New Phytologist, vol. 11, no. 2, pages 37–50, 1912.

[Jacob 04] M. Jacob & M. Unser. *Design of steerable filters for feature detection using Canny-like criteria.* IEEE TPAMI, vol. 26, no. 8, pages 1007–1019, 2004.

[Jain 90] A. K. Jain & F. Farrokhnia. *Unsupervised texture segmentation using Gabor filters.* In Systems, Man and Cybemetics, pages 14–19, 1990.

[Jain 95] S. K. Jain, M. M. Cox & R. B. Inman. *Occurrence of three-stranded DNA within a RecA protein filament.* Journal of Biological Chemistry, vol. 270, no. 9, pages 4943–4949, 1995.

[Jégou 08] H. Jégou, M. Douze & C. Schmid. *Hamming embedding and weak geometric consistency for large scale image search.* In ECCV, pages 304–317, 2008.

[Jeong 14] S.-G. Jeong, Y. Tarabalka & J. Zerubia. *Marked point process model for facial wrinkle detection.* In ICIP, pages 1391–1394, 2014.

[Jeong 15a] S.-G. Jeong, Y. Tarabalka & J. Zerubia. *Marked point process model for curvilinear structures extraction.* In EMMCVPR 2015, LNCS 8932, pages 436–449, 2015.

[Jeong 15b] S.-G. Jeong, Y. Tarabalka & J. Zerubia. *Stochastic model for curvilinear structure reconstruction using morphological profiles.* In ICIP, pages 3575–3579, 2015.

[Jiang 03] X. Jiang & D. Mojon. *Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images.* IEEE TPAMI, vol. 25, no. 1, pages 131–137, 2003.

[Joachims 99] T. Joachims. *Making large-Scale SVM Learning Practical.* In B. Schölkopf, C. Burges & A. Smola, eds., Advances in Kernel Methods - Support Vector Learning, chapitre 11, pages 169–184. MIT Press, Cambridge, MA, 1999.

[Joachims 09] T. Joachims, T. Finley & C.-N. Yu. *Cutting-plane training of structural SVMs.* Machine Learning, vol. 77, no. 1, pages 27–59, 2009.

[Kim 14] S. Kim, C. D. Yoo, S. Nowozin & P. Kohli. *Image segmentation using higher-order correlation clustering.* IEEE TPAMI, vol. 36, no. 9, pages 1761–1774, 2014.

[Kirkpatrick 83] S. Kirkpatrick, C. D. Gelatt & M. P. Vecchi. *Optimization by Simulated Annealing.* Science, vol. 220, no. 4598, pages 671–680, 1983.

[Kohavi 95] R. Kohavi. *A study of cross-validation and bootstrap for accuracy estimation and model selection.* In IJCAI, pages 1137–1143, 1995.

[Koller 95] T. M. Koller, G. Gerig, G. Szekely & D. Dettwiler. *Multiscale detection of curvilinear structures in 2-D and 3-D image data.* In ICCV, pages 864–869, 1995.

[Kumar 06] S. Kumar & M. Hebert. *Discriminative random fields.* IJCV, vol. 68, no. 2, pages 179–201, 2006.

[Lacoste 05] C. Lacoste, X. Descombes & J. Zerubia. *Point processes for unsupervised line network extraction in remote sensing.* IEEE TPAMI, vol. 27, no. 10, pages 1568–1579, 2005.

[Lafferty 01] J. Lafferty, A. McCallum & F. C. N. Pereira. *Conditional random fields: probabilistic models for segmetaing and labeling sequence data.* In ICML, pages 282–289, 2001.

[Law 08] M. W. Law & A. Chung. *Three dimensional curvilinear structure detection using optimally oriented flux.* In ECCV, pages 368–382, 2008.

[Lee 94] T. C. Lee, R. L. Kashyap & C. N. Chu. *Building skeleton models via 3-D medial surface asix thinning algorithms.* CVGIP: Graphical Models and Image Processing, vol. 56, no. 6, pages 462–478, 1994.

[Lieshout 00] M. N. M. Van Lieshout. Markov point processes and their application. Imperial College Press, 2000.

[Lindeberg 98] T. Lindeberg. *Edge detection and ridge detection with automatic scale selection.* IJCV, vol. 30, no. 2, pages 117–154, 1998.

[Lucchi 12] A. Lucchi, Y. Li, K. Smith & P. Fua. *Structured image segmentation using kernelized features.* In ECCV, pages 400–413, 2012.

[Manning 08] C. D. Manning, P. Raghavan & H. Schütze. Introduction to information retrieval. Cambridge University Press, New York, NY, USA, 2008.

[Martin 04] D. R. Martin, C. C. Fowlkes & J. Malik. *Learning to detect natural image boundaries using local brightness, color, and texture cues.* IEEE TPAMI, vol. 26, no. 5, pages 530–549, 2004.

[Mehrotra 92] R. Mehrotra, K. R. Namuduri & N. Ranganathan. *Gabor filter-based edge detection.* Pattern Recognition, vol. 25, no. 12, pages 1479–1494, 1992.

[Mittal 12] A. Mittal, M. B. Blaschko, A. Zisserman & P. H. S. Torr. *Taxonomic multi-class prediction and person layout using efficient structured ranking.* In ECCV, pages 245–258, 2012.

[Møller 03] J. Møller & R. P. Waagepetersen. Statistical inference and simulation for spatial point processes. Chapman & Hall/CRC, 2003.

[Peng 11] H. Peng, F. Long & G. Myers. *Automatic 3D neuron tracing using all-path pruning.* Bioinformatics, vol. 27, no. 13, pages 239–247, 2011.

[Perona 95] P. Perona. *Deformable kernels for early vision.* IEEE TPAMI, vol. 17, no. 5, pages 488–499, 1995.

[Philbin 07] J. Philbin, O. Chum, M. Isard, J. Sivic & A. Zisserman. *Object retrieval with large vocabularies and fast spatial matching.* In CVPR, pages 1–8, 2007.

[Platt 00] J. Platt. *Probabilistic outputs for support vector machines and comparison to regularized likelihood methods.* In A. Smola, P. Bartlett, B. Schölkopf & D. Schuurmans, eds., Advances in large margin classifiers. MIT Press, Cambridge, MA, 2000.

[Predoehl 13] A. Predoehl & K. Barnard. *A statistical model for recreational trails in aerial images.* In CVPR, pages 337–344, 2013.

[Rigamonti 12] R. Rigamonti & V. Lepetit. *Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters.* In MICCAI, pages 189–197, 2012.

[Robert 04] C. P. Robert & G. Casella. Monte Carlo statistical methods. Springer, second edition, 2004.

[Sargin 11] M. E. Sargin, A. Altinok, B. S. Manjunath & K. Rose. *Variable lenth open contour tracking using a deforamable trellis.* IEEE TIP, vol. 20, no. 4, pages 1023–1035, 2011.

[Sato 97] Y. Sato, S. Nakajima, H. Atsumi, T. Koller, G. Gerig, S. Yoshida & R. Kikinis. *3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images.* In CVRMed-MRCAS'97, LNCS 1205, pages 213–222, 1997.

[Schey 05] H. M. Schey. Div, grad, curl, and all that : an informal text on vector calculus. W.W. Norton, New York, NY, USA, 2005.

[Schlecht 07] J. Schlecht, K. Barnard, E. Spriggs & B. Pryor. *Inferring grammar-based structure models from 3D microscopy data.* In CVPR, pages 1–8, 2007.

[Sironi 14] A. Sironi, V. Lepetit & P. Fua. *Multiscale centerline detection by learning a scale-space distance transform.* In CVPR, pages 2697–2704, 2014.

[Staal 04] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever & B. van Ginneken. *Ridge based vessel segmentation in color images of the retina.* IEEE TMI, vol. 23, no. 4, pages 501–509, 2004.

[Stoyan 87] D. Stoyan, W. S. Kendall & J. Mecke. Stochastic geometry and its applications. Wiley, 1987.

[Suo 07] J. Suo, F. Min, F. Zhu, S. Shan & X. Chen. *A mutli-resolution dynamic model for face aging simulation.* In CVPR, pages 1–8, 2007.

[Sziranyi 14] T. Sziranyi & M. Shadaydeh. *Segmentation of remote sensing images using similarity-measure-based fusion-MRF mode.* IEEE GRSL, vol. 11, no. 9, pages 1544–1548, 2014.

[Talbot 07] H. Talbot & B. Appleton. *Efficient complete and incomplete path openings and closings.* Image and Vision Computing, vol. 25, no. 4, pages 416–425, 2007.

[Taskar 03] B. Taskar, C. Guestrin & D. Koller. *Max-margin Markov networks.* In NIPS, pages 25–32, 2003.

[Tsochantaridis 04] I. Tsochantaridis, T. Joachims, T. Hofmann & Y. Altun. *Support vector machine learning for interdepedent and structured output space.* In ICML, pages 104–112, 2004.

[Tsochantaridis 05] I. Tsochantaridis, T. Joachims, T. Hofmann & Y. Altun. *Large margin methods for structured and interdependent output variables.* Journal of Machine Learning Research (JMLR), vol. 6, pages 1453–1484, 2005.

[Tu 06] Z. Tu & S.-C. Zhu. *Parsing images into regions, curves, and curve groups.* IJCV, vol. 69, no. 2, pages 223–249, 2006.

[Türetken 11] E. Türetken, G. González, C. Blum & P. Fua. *Automated reconstruction of dendritic and axonal tress by global optimization with geometric priors.* Neuroinformatics, vol. 9, no. 2–3, pages 279–302, 2011.

[Türetken 13a] E. Türetken, C. Becker, P. Glowacki, F. Benmansour & P. Fua. *Detecting irregular curvilinear structures in gray scale and color imagery using multi-directional oriented flux.* In ICCV, pages 1553–1560, 2013.

[Türetken 13b] E. Türetken, F. Benmansour, B. Andres, H. Pfister & P. Fua. *Reconstructing loopy curvilinear structures using integer programming.* In CVPR, pages 1822–1829, 2013.

[Unsalan 12] C. Unsalan & B. Sirmacek. *Road network detection using probabilistic and graph theoretical methods.* IEEE TGRS, vol. 50, no. 11, pages 4441–4453, 2012.

[Valero 10] S. Valero, J. Chanussot, J. A. Bendiktsson, H. Talbot & B. Waske. *Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images.* Pattern Recognition Lett., vol. 31, no. 10, pages 1120–1127, 2010.

[Vapnik 95] V. Vapnik. The nature of statistical learning theory. Springer Series in Information Science and Statistics, Springer New York, NY, USA, 1995.

[Vasilevskiy 01] A. Vasilevskiy & K. Siddiqi. *Flux maximizing geometric flows.* IEEE TPAMI, vol. 24, no. 12, pages 1565–1578, 2001.

[Verdié 14] Y. Verdié & F. Lafarge. *Detecting parametric objects in large scenes by Monte Carlo sampling.* IJCV, vol. 106, no. 1, pages 57–75, 2014.

[Vosselman 95] G. Vosselman & J. de Knecht. *Road tracing by profile matching and Kalman filtering.* In Automatic Extraction of Man-Made Objects from Aerial and Sapce Images, pages 265–274, 1995.

[Wang 11] Y. Wang, A. Narayanaswamy & B. Roysam. *Novel 4D open-curve active contour and curve completion approach for automated tree structure extraction.* In CVPR, pages 1105–1112, 2011.

[Wick 11] M. Wick, K. Rohanimanesh, K. Bellare, A. Culotta, & A. McCallum. *SampleRank: training factor graphs with atomic gradients.* In ICML, pages 777–784, 2011.

[Yang 11] Y. Yang & D. Ramanan. *Articulated pose estimation using flexible mixtures of parts.* In CVPR, pages 1386–1392, 2011.

[Zana 01] F. Zana & J. Klein. *Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation.* IEEE TIP, vol. 10, no. 7, pages 1010–1019, 2001.

[Zhao 11] T. Zhao, J. Xie, F. Amat, N. Clack, P. Ahammad, H. Peng, F. Long & E. Myers. *Automated reconstruction of neuronal morphology based on local geometrical and global structural models.* Neuroinformatics, vol. 9, no. 2–3, pages 247–261, 2011.

[Zheng 07] Z. Zheng, H. Zha, T. Zhang, O. Chapelle & G. Sun. *A general boosting method and its application to learning ranking functions for web search.* In NIPS, pages 1697–1704, 2007.